

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2000-200207  
(P2000-200207A)

(43) 公開日 平成12年7月18日 (2000.7.18)

(51) Int.Cl.<sup>7</sup>  
G 0 6 F 12/00

識別記号  
5 2 0

F I  
G 0 6 F 12/00

テーマコード (参考)  
5 B 0 8 2

審査請求 未請求 請求項の数15 O L (全 31 頁)

(21) 出願番号 特願平10-343433

(22) 出願日 平成10年12月2日 (1998.12.2)

(31) 優先権主張番号 特願平10-311182

(32) 優先日 平成10年10月30日 (1998.10.30)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 五十嵐 卓也

東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内

Fターム (参考) 5B082 EA01

(54) 【発明の名称】 階層管理ファイル装置及びこれを具備した電子機器

(57) 【要約】

【課題】 連続且つ高速記録再生を要する大サイズのフ  
ァイルと離散的且つ小サイズのファイルを混在して効率  
よく記録再生できる階層管理ファイル装置及びこれを具  
備した電子機器を提供する。

【解決手段】 情報記録媒体の記録領域を領域602～  
610のクラスタに分割し、各クラスタを領域611～  
618で示すフラグメントにより再分割し、各クラスタ  
の連鎖位置関係とクラスタに属するフラグメントの使用  
状況を示すクラスタFATと、各フラグメントの連結情  
報を示すフラグメントFATと、クラスタFATとフラ  
グメントFATを統括するディレクトリエントリテーブ  
ルとを参照して、情報記録媒体に階層的にファイルを記  
録再生する。

601 システム領域
602 クラスタ 2
603 3
604 4
605 5
606 6
607 7
608 8
609 N-1
610 N

ボリューム構造

611 フラグメント0
612 1
613 3
614 4
615 5
616 6
617 126
618 127

クラスタ構造

## 【特許請求の範囲】

【請求項1】 情報記録媒体の記録領域を複数種の記録単位長により階層的に分割し、各階層の分割領域が、下位階層の複数の分割領域で構成され、ファイルを前記情報記録媒体に前記各階層の分割領域の単位で、第1の管理情報及び第2の管理情報により階層管理してファイル記録領域に分割記録し、前記各階層の分割領域を前記第1の管理情報及び前記第2の管理情報により階層管理して前記ファイルを前記情報記録媒体より再生する階層管理ファイル装置において、

前記各階層の分割領域の連結情報と使用情報とを階層別に有する前記第1の管理情報と、前記ファイルの属性、該ファイルの記録再生に使用される階層の種類及びファイルの種類、及び記録再生の始まる最初の階層の分割領域の位置を前記ファイル毎に有する前記第2の管理情報とが記録再生される前記情報記録媒体と、各階層の前記第1の管理情報及び前記第2の管理情報を参照して、前記ファイルを前記各階層の分割領域に記録する場合は、前記下位階層の第1の管理情報を参照して、該下位階層の前記分割領域の割り当てを行い、空いている分割領域がない場合は、上位階層の前記第1の管理情報を参照し、再度、当該上位階層に属する前記下位階層の前記第1の管理情報を検索し、前記ファイルの前記分割領域への割り当てを可能とし、前記ファイルの特性に応じて階層の種類を使い分けて前記情報記録媒体上に、前記ファイルを記録再生する制御を行うソフトウェア制御手段を具備したことを特徴とする階層管理ファイル装置。

【請求項2】 前記ソフトウェア制御手段は、前記第1の管理情報と、前記第2の管理情報と、前記ファイル記録領域とを、階層別に昇順番号を付された分割領域に記録再生し、該昇順番号により前記分割領域を識別し得ることを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項3】 前記ソフトウェア制御手段は、前記第2の管理情報が、同一の階層の分割領域に記録再生され、該第2の管理情報が、少なくとも、前記ファイルの属性を示す第1の情報と、前記ファイルの記録再生に使用される前記階層の種類及びファイルの種類を示す第2の情報と、記録再生の始まる最初の階層の分割領域の位置を示す第3の情報とで構成され、前記ファイルを前記第2の情報により階層ファイルでないと識別するとき、前記第3の情報と識別された階層の前記第1の管理情報とを参照して、当該ファイルを記録再生し得ること特徴とする請求項1に記載の階層管理ファイル装置。

【請求項4】 前記ソフトウェア制御手段は、前記第2の管理情報が、同一の階層の分割領域に記録再生され、該第2の管理情報が、少なくとも、前記ファイルの属性を示す第1の情報と、前記ファイルの記録再生に使用される前記階層の種類及びファイルの種類を示す第2の情

報と、記録再生の始まる最初の階層の分割領域の位置を示す第3の情報とで構成され、前記ファイルが前記第2の情報により、親ファイルと子ファイルで構成される階層ファイルであると識別したとき、前記親ファイルの階層の前記分割領域の最初の位置を示す前記第3の情報と一致する前記子ファイルの第2の情報を検索し該子ファイルを識別し、該子ファイルの前記第3の情報により、該子ファイルの記録再生の始まる前記最初の階層の分割領域の位置を識別し、識別した該子ファイルの階層の前記第1の管理情報を参照して、子ファイルの使用する前記下位階層の分割領域は必ず前記親ファイルの使用する前記上位階層の分割領域に含まれるように、前記ファイルを記録再生し得ることを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項5】 前記ソフトウェア制御手段は、前記第1の管理情報が当該階層の分割領域の前記連結情報及び使用情報を示す第4の情報と前記下位階層の分割領域の使用状況を示す第5の情報で構成され、前記第4の情報により、前記当該階層の任意の前記分割領域の次に前記ファイルを記録再生すべき前記分割領域を識別して、前記ファイルに割り当てられた前記分割領域を連続的に記録再生でき、且つ前記第5の情報により前記下位階層の分割領域を前記ファイルの記録領域に割り当てるか否かを制御し得ることを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項6】 前記下位階層に属する前記第1の管理情報は当該下位階層の分割領域間の前記連結情報及び使用情報と、当該下位階層の属する前記上位階層の分割領域間の前記連結情報及び使用情報の情報を有することを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項7】 前記第1の管理情報中に前記ファイルの終了を示す情報を階層別に有することを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項8】 前記ソフトウェア制御手段が、前記情報記録媒体の初期化において、最大の前記記録単位長により分割された最上位階層の前記分割領域のみを割り当てることを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項9】 前記ソフトウェア制御手段が、前記下位階層の分割領域に前記ファイルの連続した複数の情報単位を割り当て得ることを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項10】 前記ソフトウェア制御手段は、新規の前記分割領域をファイル用に確保した場合及び該新規の前記分割領域に前記ファイルが記録されたときは、前記第1の管理情報中の、当該階層の分割領域の前記連結情報及び使用情報を示す第4の情報と下位階層の分割領域の使用状況を示す第5の情報とを階層別に更新することを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項11】 前記ソフトウェア制御手段は、新規の

前記分割領域を前記ファイル用に確保した場合及び該新規の分割領域に前記ファイルを記録したときは、前記ファイルの属性を示す第1の情報と、前記第2の管理情報中の、前記ファイルの記録再生に使用される前記階層の種類及び前記ファイルの種類を示す第2の情報と、記録再生の始まる最初の階層の分割領域の位置を示す第3の情報とを更新することを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項12】 前記ソフトウェア制御手段は、前記ファイルの記録において、前記第1の管理情報と前記第2の管理情報を新規の分割領域を割り当てて更新する場合、該第1の管理情報と該第2の管理情報を参照し、該ファイルが記録された最新の階層の前記分割領域の位置を基準として、該最新の階層の前記分割領域が属する最上位階層の分割領域の前記第1の管理情報を参照し、未使用の前記最上位階層の分割領域を前記情報記録媒体全体にわたり検索し、次々に階層をさげ、前記未使用の前記最上位階層の分割領域の検索と同様の検索を行い、前記最新の階層の分割領域の位置に最も近い前記分割領域に前記ファイルの前記新規に割り当てられる情報を記録することを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項13】 前記ソフトウェア制御手段は、階層ファイルの記録において、前記第1の管理情報と前記第2の管理情報を新規の分割領域を割り当てて更新する場合、該第1の管理情報と該第2の管理情報を参照し、前記ファイルが記録された最新の階層の前記分割領域の位置を基準として、該最新の階層の分割領域が属する前記最上位階層の分割領域の中、親ファイルが属する前記第1の管理情報を参照し、前記親ファイルに属する未使用の最上位階層の分割領域を前記情報記録媒体全体にわたり検索し、次々に階層をさげ、前記親ファイルに属する未使用の最上位階層の分割領域の検索と同様の検索を行い、前記最新の階層の分割領域の位置に最も近い分割領域に前記階層ファイルの前記新規に割り当てられる子ファイルの情報を記録することを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項14】 前記第1の管理情報が、前記各階層の分割領域の使用情報と前記下位階層の分割領域の使用状況を階層別にもち、前記各階層の分割領域の連結情報を、前記第2の管理情報、及び前記第1の管理情報と前記第2の管理情報とは別の記録領域に設けられた第3の管理情報の何れか一方に有することを特徴とする請求項1に記載の階層管理ファイル装置。

【請求項15】 請求項1に記載の階層管理ファイル装置を具備したことを特徴とする電子機器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は階層管理ファイル装置及びこれを具備した電子機器に関し、さらに詳しく

は、情報記録媒体の記録領域を複数種の記録単位長により階層的に分割し、各階層の分割領域が、下位の階層の複数の分割領域で構成され、ファイルを情報記録媒体に、各階層の分割領域の単位で管理情報により階層管理してファイル記録領域に分割して記録し、当該情報記録媒体より、各階層の分割領域を管理情報により階層管理してファイルを再生する階層管理ファイル装置及びこれを具備した電子機器に関する。

【0002】

【従来の技術】 ハードディスクや光ディスク等、2次記録装置を持つ情報処理装置では記録装置の記録領域をファイルとして読み書きできるように管理するファイル装置という機構が実現されている。古くから大型計算機等でも様々なファイル装置が考案、実用化されている。が、79年頃から出現したパーソナルコンピュータと言われる小型のコンピュータ機器でもフロッピーディスクやハードディスクを管理するファイル装置が使用され、FAT (File Allocation Table) と呼ばれるファイル配置管理テーブルを用いてこれを実現している。FATはコンピュータ機器のみならず民生用産業用機器のファイル装置としても採用され、その構造が簡単であるため、パーソナルコンピュータや民生産業用機器のオペレーティングシステムに組み込まれ使用されている。

【0003】 図33はパーソナルコンピュータに用いられる磁気ディスクを示したものである。ブートセクタはオペレーティングシステムを起動するためのロードプログラムや後述するルートディレクトリの番号、前述したFATの番号などが記録される領域である。また、ルートディレクトリの領域には、記録されているファイルのファイル名、ファイルサイズなどが記録される領域である。FATは前述したようにファイル配置管理テーブルであり、所望のファイルはルートディレクトリとFATにより記録領域から読み出したり、書き込んだりすることができる。

【0004】 図34、図35、図36、を参照して、FATによるファイル装置の概要を説明する。図34に示すように、情報記録媒体の記録領域（以下ボリュームという）は主記録単位長（以下クラスタという）で分割され、各々のクラスタにはクラスタ番号が割り振られており、情報はクラスタの単位で分割され記録、再生が行われる。

【0005】 ここで、以下に説明を続けるにあたり、後述するクラスタ番号と、単に図面の説明を行う上での符号とを混同しないように記載方法を明確にしておく。すなわち、例えば、領域300のように記載した場合、領域とは、一つのクラスタの記録される領域、複数のクラスタが記録される記録領域、あるいは、クラスタの記録領域の部分を意味し、且つ、図面によりこれら記録領域を示し、数値300は、単に、図面上での上述の記録領域を示す符号を意味する。また、FATエントリー30

0と記載するときは、領域300がFATの構成単位であるエントリーの一つを構成していることを意味する。

【0006】クラスタは通常512バイトから32キロバイトのサイズになっている。ここでは512バイトとする。例えば領域102のクラスタのクラスタ番号は2、領域103のクラスタは3となる。また、領域101のシステム領域には記録領域全体情報が記録されており、総クラスタ数やクラスタのサイズを知ることができる。

【0007】ファイル装置ではディレクトリと呼ばれる管理領域にて複数のファイルを一まとめにして管理し、そのディレクトリに属する各々のファイルの情報、例えばファイル名、作成日、ファイルサイズなどの情報を記録している。

【0008】図35にディレクトリの構造を示す。これらディレクトリはファイルと同様にクラスタの単位で記録される。ディレクトリはルートディレクトリからサブディレクトリを参照されることで階層的になっており、階層的にディレクトリをたどることにより、目的のファイルのディレクトリエントリに到達する。

【0009】図35の領域204のFILE2.DATはファイル名であり、このファイルのファイルサイズは領域206のファイルサイズで示される通り2013バイトであり、さらに領域205にはファイルを構成するクラスタの最初のクラスタの番号4が記録されている。このクラスタ番号から次々にFATのチェーンをたどることによりファイルを構成するすべてのクラスタに対して読み書きを行うことができる。

【0010】図36にはFATの構造を示す。FATの各エントリは図34に示された記録領域の各クラスタに対応しており、図36の(2)～(12)で示したクラスタ番号の順に並んでいる。各FATのエントリにはそのファイルを構成する次のクラスタの番号が記録されている。例えば、図35のディレクトリエントリより示されたFILE2.DATの最初のクラスタ4(クラスタ番号が(4)のクラスタの意味、以下同様)のFATエントリ305の値は7となっており、次のクラスタは(7)であることがわかる。次にクラスタ7のFATエントリ308の値は11であり、クラスタ(11)のFATエントリ312の値は9であり、さらにクラスタ(9)のFATエントリ310の値は65535であり、65535は特別な意味をもち、最終のクラスタであることを示す。

【0011】これらFATエントリのチェーンにより、FILE2.DATは図34で示されるディスク記録領域104のクラスタ4、領域107のクラスタ7、領域111のクラスタ11、領域109のクラスタ9の4つのクラスタで構成されていることがわかる。一方、領域206によるとファイルサイズは2013であるから、この場合は1つのクラスタは512バイトであり、最終

のクラスタは477バイトのみ使用されていることになる(2013-512×3=477)。また、領域306、領域311、領域313のFATエントリの値は0となっており、この値はクラスタが未使用なことを示し、それに対応する領域105、領域110、領域112のクラスタ5、クラスタ10、クラスタ12は、ファイルのサイズが増えた場合、新規のファイルの場合の記録領域として使用される。

【0012】

【発明が解決しようとする課題】このようにFATによる管理は構造が簡単であるが、その一方で以下の3つの問題がある。

(1) クラスタが連続して使用されている場合でも、FATエントリを個々に参照しなければならないため、クラスタが多い場合には、FATエントリの参照回数が増え、ファイルをランダムアクセスする場合など効率が悪い。

【0013】(2) ファイルアクセスを高速に行うにはできるだけ連続したクラスタにファイルのデータを配置する方が良いが、空きクラスタも個々のFATエントリに対応するために、連続した空きクラスタを探すために個々の空きクラスタを参照することになり、FATエントリの参照回数が増えること、及びファイルを構成するエクステン(連続したクラスタの単位)ができるだけ分断(フラグメンテーション)されないように連続クラスタに配置するには効率が悪い。

【0014】(3) 記録装置の大容量化に伴いFATのサイズ自体が大きくなり、FATエントリを参照するためのCPUの処理速度、メモリ容量を多く必要とし、このため、フラグメンテーションが起きない様にファイルの配置を行おうとしてもハードウェアの使用効率が悪い。

【0015】(4) 技術の進歩によりコンピュータ機器の扱うデータも多様になっており、ワードプロセッサの文章など数キロバイトのファイルから、オーディオ、ビデオのデータを含んだ数10メガ、数メガ、さらには数ギガの大きさのファイルまで同一の情報記録媒体に記録する必要がでてきている。様々なサイズのファイルが存在することはフラグメンテーションを起きやすくし、クラスタのサイズを小さくすることによって記録領域を有効に使用することができるが、ファイルが分断されて配置された場合はデータの転送速度が低下するため、AVデータの再生、録音をリアルタイムでできないなどの影響を与える。

【0016】図37にフラグメンテーションによるデータ転送の低下の例を示す。領域401、領域403、405、407のクラスタは分断されて配置されているため、データ読み出しの他に、領域402、404、406で示されるシークの時間が必要とされ、このシーク時間がデータ転送の低下を引き起こす。

【0017】従来FATのエントリを32ビットにして多くのクラスタを管理できる拡張されたFATを実装して、記録装置の大容量化に対応をしているが、上記

(1)、(2)、(3)、(4)の問題に関しては不十分であり、また、クラスタのサイズを小さくすることにより、情報記録媒体の記録領域を有効に使う工夫もなされているが、逆に、この場合はAVファイル等大きなサイズのファイルはフラグメンテーションは起きやすくなる。そこで、フラグメンテーションに関しては、いわゆるディフラグメンテーションというソフトウェアにて、

10 ファイルを記録後クラスタのデータを再配置、再記録することによって対処している。

【0018】本発明はこのような状況を鑑みてなされたものであり、簡易なFATの管理方法を採用しつつ、大きな割り当て単位のクラスタFATと、そのクラスタをさらに分割した小さな割り当て単位のフラグメントFATで階層的に記録領域を管理することにより上記

(1)、(2)、(3)、(4)の問題を解決し、フラグメンテーションを効率的に防ぐことができ、さまざまなサイズのファイルを効率良く記録でき、AVデータな

20 どのリアルタイム性を保証する階層管理ファイル装置及びこれを具備した電子機器を提供することを課題とする。

【0019】

【課題を解決するための手段】課題を達成するために、本発明の階層管理ファイル装置及びこれを具備した電子機器では、情報記録媒体の記録領域を複数種の記録単位長により階層的に分割し、各階層の分割領域が、下位階層の複数の分割領域で構成され、ファイルを情報記録媒体に各階層の分割領域の単位で、第1の管理情報及び第

2の管理情報により階層管理してファイル記録領域に分割記録し、各階層の分割領域を第1の管理情報及び第2の管理情報により階層管理してファイルを情報記録媒体より再生する階層管理ファイル装置において、各階層の分割領域の連結情報と使用情報とを階層別に有する第1の管理情報と、ファイルの属性、ファイルの記録再生に使用される階層の種類及びファイルの種類、及び記録再生の始まる最初の階層の分割領域の位置をファイル毎に有する第2の管理情報とが記録再生される情報記録媒体と、各階層の第1の管理情報及び第2の管理情報を参照して、ファイルを各階層の分割領域に記録する場合は、下位階層の第1の管理情報を参照して、下位階層の分割領域の割り当てを行い、空いている分割領域がない場合は、上位階層の第1の管理情報を参照し、再度、当該上位階層に属する下位階層の第1の管理情報を検索し、ファイルの分割領域への割り当てを可能とし、ファイルの特性に応じて階層の種類を使い分けて情報記録媒体上に、ファイルを記録再生する制御を行うソフトウェア制御手段を具備したことを特徴とする。

【0020】また、本発明の階層管理ファイル装置の望

ましい形態としては下記(1)～(13)である。

(1) ソフトウェア制御手段は、第1の管理情報と、第2の管理情報と、ファイル記録領域とを、階層別に昇順番号を付された分割領域に記録再生し、該昇順番号により分割領域を識別し得るものである。

【0021】(2) ソフトウェア制御手段は、第2の管理情報が、同一の階層の分割領域に記録再生され、第2の管理情報が、少なくとも、ファイルの属性を示す第1の情報と、ファイルの記録再生に使用される階層の種類及びファイルの種類を示す第2の情報と、記録再生の始まる最初の階層の分割領域の位置を示す第3の情報とで構成され、ファイルを前記第2の情報により階層ファイルでないと識別するとき、第3の情報と識別された階層の第1の管理情報とを参照して、当該ファイルを記録再生し得るものである。

【0022】(3) ソフトウェア制御手段は、第2の管理情報が、同一の階層の分割領域に記録再生され、第2の管理情報が、少なくとも、ファイルの属性を示す第1の情報と、ファイルの記録再生に使用される階層の種類及びファイルの種類を示す第2の情報と、記録再生の始まる最初の階層の分割領域の位置を示す第3の情報とで構成され、ファイルが第2の情報により、親ファイルと子ファイルで構成される階層ファイルであると識別したとき、親ファイルの階層の分割領域の最初の位置を示す第3の情報と一致する子ファイルの第2の情報を検索し子ファイルを識別し、子ファイルの第3の情報により、子ファイルの記録再生の始まる最初の階層の分割領域の位置を識別し、識別した該子ファイルの階層の第1の管理情報を参照して、子ファイルの使用する下位階層の分割領域は必ず親ファイルの使用する上位階層の分割領域に含まれるように、ファイルを記録再生し得るものである。

【0023】(4) ソフトウェア制御手段は、第1の管理情報が当該階層の分割領域の前記連結情報及び使用情報を示す第4の情報と下位階層の分割領域の使用状況を示す第5の情報で構成され、第4の情報により、当該階層の任意の分割領域の次にファイルを記録再生すべき分割領域を識別して、ファイルに割り当てられた分割領域を連続的に記録再生でき、且つ第5の情報により下位階層の分割領域をファイルの記録領域に割り当てるか否かを制御し得るものである。

【0024】(5) 下位階層に属する第1の管理情報は当該下位階層の分割領域間の連結情報及び使用情報と、当該下位階層の属する上位階層の分割領域間の連結情報及び使用情報の情報を有するものである。

【0025】(6) 第1の管理情報中にファイルの終了を示す情報を階層別に有するものである。

【0026】(7) ソフトウェア制御手段が、情報記録媒体の初期化において、最大の記録単位長により分割された最上位階層の分割領域のみを割り当てるものであ

る。

【0027】(8) ソフトウェア制御手段が、下位階層の分割領域に前記ファイルの連続した複数の情報単位を割り当て得るものである。

【0028】(9) ソフトウェア制御手段は、新規の分割領域をファイル用に確保した場合及び新規の分割領域にファイルが記録されたときは、第1の管理情報中の、当該階層の分割領域の連結情報及び使用情報を示す第4の情報と下位階層の分割領域の使用状況を示す第5の情報とを階層別に更新するものである。

【0029】(10) ソフトウェア制御手段は、新規の分割領域をファイル用に確保した場合及び新規の分割領域にファイルを記録したときは、ファイルの属性を示す第1の情報と、第2の管理情報中の、ファイルの記録再生に使用される階層の種類及びファイルの種類を示す第2の情報と、記録再生の始まる最初の階層の分割領域の位置を示す第3の情報とを更新するものである。

【0030】(11) ソフトウェア制御手段は、ファイルの記録において、第1の管理情報と第2の管理情報を新規の分割領域を割り当てて更新する場合、第1の管理情報と第2の管理情報を参照し、ファイルが記録された最新の階層の分割領域の位置を基準として、最新の階層の分割領域が属する最上位階層の分割領域の第1の管理情報を参照し、未使用の最上位階層の分割領域を情報記録媒体全体にわたり検索し、次々に階層をさげ、未使用の最上位階層の分割領域の検索と同様の検索を行い、最新の階層の分割領域の位置に最も近い分割領域にファイルの新規に割り当てられる情報を記録するものである。

【0031】(12) ソフトウェア制御手段は、階層ファイルの記録において、第1の管理情報と第2の管理情報を新規の分割領域を割り当てて更新する場合、第1の管理情報と第2の管理情報を参照し、ファイルが記録された最新の階層の分割領域の位置を基準として、最新の階層の分割領域が属する最上位階層の分割領域の中、親ファイルが属する第1の管理情報を参照し、親ファイルに属する未使用の最上位階層の分割領域を情報記録媒体全体にわたり検索し、次々に階層をさげ、親ファイルに属する未使用の最上位階層の分割領域の検索と同様の検索を行い、最新の階層の分割領域の位置に最も近い分割領域に階層ファイルの新規に割り当てられる子ファイルの情報を記録するものである。

【0032】(13) 第1の管理情報が、各階層の分割領域の使用情報と下位階層の分割領域の使用状況を階層別にもち、各階層の分割領域の連結情報を、第2の管理情報、及び第1の管理情報と前記第2の管理情報とは別の記録領域に設けられた第3の管理情報の何れか一方に有するものである。

【0033】前述した作用について以下(1)～(3)までに記す。

(1) 情報記録媒体の記録領域を複数の記録単位長によ

り階層的に分割し、各階層の分割領域が、下位階層の複数の分割領域で構成され、第1の管理情報と第2の管理情報を参照してソフトウェア制御手段によりファイルを記録再生できるため、例えば、映像や音声のような連続して再生する必要のあるファイルには大きな記録単位長で分割された分割領域を割り当てることにより、分割領域の数を少なくでき、且つ第1の管理情報の参照回数を少なくできる。また、文章のような情報量が小さく、離散的に再生してもよいファイルは小さな記録単位長で分割された分割領域を割り当てることができ情報記録媒体の記録領域を有効に利用することができる。

【0034】(2) ファイルのアクセスを高速に行うにはできるだけ連続した分割領域にファイルの個々の情報を配置する方がよい。情報記録媒体の記録領域を複数の記録単位長により階層的に分割し、各階層の分割領域が、下位階層の複数の分割領域で構成され、上位階層の分割領域が下位階層の分割領域の使用状況を示す情報を有していることにより、ファイルの分割領域への割り当てに際して、未使用の分割領域を階層的に検索することができ、フラグメンテーションの発生を抑え、まとまった分割領域にファイルを高速で記録することができ、且つ記録されたファイルの再生も高速に行うことができる。

【0035】(3) 映像や音声のような連続して再生する必要のあるファイルには大きな記録単位長で分割された分割領域を割り当て、文章のような情報量が小さく、離散的に再生してもよいファイルは小さな記録単位長で分割された分割領域を割り当てることにより、第1の管理情報の情報量も格段少なくでき、また、(1)及び(2)で述べたことも併せて、本発明の階層管理ファイル装置を具備した電子機器に使用するハードウェア装置の負荷を軽減することができる。

【0036】

【発明の実施の形態】以下では、本発明の階層管理ファイル装置及びこれを具備した電子機器の一例として、前述のクラスタを上位階層の分割領域とし、後述するフラグメントを下位階層の分割領域とし、上位階層の第1の管理情報としてクラスタFATを用い、下位階層の第1の管理情報としてフラグメントFATを用い、第2の管理情報として、クラスタで構成されるディレクトリエントリテーブルを用いた、2階層の階層管理ファイル装置及びこれを具備した電子機器について説明するが、本発明はこれに限るものではなく、一般の複数階層の階層管理ファイル装置及びこれを具備した電子機器に適用可能である。

【0037】図1に本発明の階層管理ファイル装置及びこれを具備した電子機器の実施の形態例の概略構成を示す。CPU501はメモリ502に記録されたプログラムに従ってシステムの制御を行う。メモリ502としては高速の半導体メモリを使用してもよい。プログラムは

あらかじめ、図を省略するROM (Read Only Memory) に記録されていてもかまわないが、2次記録装置503、ネットワークドライバ504を介して、また、チューナ509などからメモリ502にプログラムを転送して実行しても構わない。2次記録装置503はハードディスクや交換可能な光ディスクまたはFlash ROMなどの不揮発性メモリが用いられる。2次記録装置503中の情報記録媒体に記憶されたファイルはメモリ502に保存されたプログラムに従って本発明の階層管理ファイル装置510により管理される。請求項1に記載のソフトウェア制御手段はCPU501、メモリ502で構成される。本発明の階層管理ファイル装置510は、2次記録装置503、ソフトウェア制御手段及び情報記録媒体で構成される。

【0038】ネットワークドライバ504は符号505で示したIEEE1394やEthernetなどのネットワークをサポートし、2次記録装置503に記録されるファイルのデータの転送または2次記録装置503から読み出されたファイルデータの転送に用いられる。また、これらデータは一旦、メモリ502のプログラムに従ってCPU501またはバス506に接続された特殊なハードウェアにより圧縮、伸張などのデータ加工をした後、ネットワークへの転送をすることも可能である。

【0039】赤外線送受信装置507はユーザからのリモコン操作によりコマンドを受信してメモリ502に記録されたプログラムの動作を制御して対話的な動作を行うことが可能である。画像音声入出力装置508はテレビまたはモニターなどの表示装置に接続され、メモリ502に保存されたプログラムに従って、グラフィカルユーザインターフェース (図示せず) や2次記録装置503、ネットワークドライバ504から転送された画像、音声を出力する。

【0040】また、画像音声入出力装置508はビデオカメラやビデオデッキなどの機器と接続され、ビデオデッキからの音声映像の取り込みおよびビデオデッキへの記録などが行える。チューナ509は衛星放送のアンテナや地上波テレビ放送のアンテナと接続され、受信したデータは画像音声入出力装置508に送られ再生される。

【0041】また、これらデータは2次記録装置503に保存したのち、再生することも可能である。画像音声入出力装置508、チューナ509が扱う入出力信号はアナログでもデジタルでも構わないが、アナログの場合は画像音声入出力装置508、チューナ509においてデジタルデータに変換され、デジタルデータは圧縮、データ変換などが行われ、図1の本発明の階層管理ファイル装置及びこれを具備した電子機器 (以下システム構成という) の中で処理される。

【0042】CPU501、メモリ502、2次記録装

置503、ネットワークドライバ504、赤外線送受信装置507、画像音声入出力装置508、チューナ509はバス506に接続され、システムを構成する。なお、図1のシステム構成は一例であって、必要に応じてネットワークドライバ504、赤外線送受信装置507、映像音声入出力装置508、チューナ509の中、何れかがないシステム構成も可能である。例えば、2次記録装置503もなしで、ネットワークドライバ504などを介して遠隔のシステムに接続された2次記録装置503にファイル装置により記録することも可能である。また、用途に合わせて他のデバイスが接続されたシステム構成も取りうる。

【0043】図2に本発明の階層管理ファイル装置510によるボリューム管理の方法を示す。従来のFATによるファイル装置と同様に、2次記録装置503の記録領域であるボリュームはあるクラスタ (固定サイズ) 単位に分割して管理され、各々のクラスタには連続したクラスタ番号 (例では2以上の整数) が割り振られる。

【0044】さらに本発明においてはクラスタは、さらに小さなサイズのフラグメント単位に分割して管理され、クラスタ内のフラグメントには連続したフラグメント番号が割り振られる。例ではクラスタのサイズを64キロバイトとし、フラグメントのサイズを512バイトとし、クラスタは128個のフラグメントに分割され、フラグメント番号が0から127まで各々のフラグメントに割り振られている。

【0045】本発明の階層管理ファイル装置510ではクラスタを単位としたクラスタFATと、フラグメントを単位としたフラグメントFATが用いられる。クラスタFATにはボリュームの全クラスタ数のFATエントリが記録される。フラグメントFATには、全クラスタ数とクラスタ内のフラグメントの数の積の数だけのFATエントリが記録される。

【0046】FATは前述したように、クラスタ番号よりそれに対応したクラスタエントリを容易に知ることができる構造を取る。FATは2次記録装置503中のクラスタに記録され、システム領域からその場所 (クラスタ番号) が示されている。効率化のため、FAT (1部) はメモリ502に読み込まれ、必要に応じて2次記録装置503に再度、書き込まれ更新される。

【0047】図3にクラスタFATのエントリ形式を示す。クラスタFATのエントリには、対応するクラスタの使用方法によって、以下の6つの形式で32ビットの値が記録される。

- 1) 未使用のクラスタ
- 2) フラグメント単位で使用中のクラスタ
- 3) フラグメント単位で使用中のクラスタ (すべてのフラグメントを使用中)
- 4) ファイルに割り当て済みのクラスタ
- 5) ファイルに割り当て済みのクラスタ (階層ファイル

にてフラグメント単位で使用中)

【0048】1)の未使用のクラスタの場合は、領域701の上位25ビットに識別子0の値が記録される。このクラスタは使用されておらず、2)から5)の形式に割り当て可能であることを示す。ボリュームの初期化時にはすべてのクラスタFATのエントリは未使用の状態に初期化される。

【0049】2)のフラグメント単位で使用中の場合は、領域702の上位25ビットは識別子0の値が記録され、領域703の下位7ビットにはそのクラスタ中の使用中のフラグメント数が記録される。使用中フラグメントの数により、フラグメントFATを参照することなしに、そのクラスタの使用状況が判断でき、その情報をもとに効率良くフラグメントをファイルに割り当てることができる。たとえば、割り当てに必要なフラグメントが多ければ、できるだけ使用中フラグメントが多いクラスタを使用することにより、ファイルのフラグメントをまとめた領域に記録することができ、データのアクセスを高速化できる。

【0050】3)の場合は、2)の場合で、且つ、すべてのフラグメントが使用されていた場合であり、領域704の上位25ビットには識別子1の値が記録され、下位7ビットには識別子0が記録される。

【0051】2)、3)の場合ともファイルはフラグメントを単位としてボリュームの記録領域が割り当てられ、ファイルを構成するすべてのフラグメントはフラグメントFATのFATエントリにより連結されている。

【0052】4)のファイルに割り当て済みのクラスタの場合は、領域706の上位25ビットにファイルを構成する次のクラスタのクラスタ番号が記録され、領域707の下位7ビットには識別子0が記録される。また、4)の場合で、且つ、次のクラスタがない(つまりファイルの最終クラスタ)場合は、領域706の上位25ビットには、例えば、識別子33、554、431が記録される。

【0053】4)の場合、ファイルはクラスタを単位としてボリュームの記録領域が割り当てられており、ファイルを構成するすべてのクラスタはクラスタFATのFATエントリにより連結されている。

【0054】5)の場合は、本発明の階層管理ファイル装置510の適用効果を最もよく示すものであり、ファイルはクラスタを単位にボリュームの記録領域が割り当てられるが、その割り当てられたクラスタはフラグメントに分割して、さらに、他のファイルに割り当てられる。これを階層ファイルといい、クラスタ単位で管理されるファイルを親ファイル、その親ファイルのクラスタ中のフラグメントを単位として管理されるファイルを子ファイルと呼ぶ。この階層ファイルを使用することにより、子ファイルをまとめた記録領域に効率良く記録することができ、複数の子ファイルを連続して読み出し書

きこみする場合などで、アクセス時間を短縮することができる。

【0055】また、フラグメントの割り当て方法、たとえば、連続する4つのフラグメントを単位にファイルに割り当てるようにクラスタを使用し、そのクラスタのまとまりを親ファイルとして管理するなどの方法も取ることができる。

【0056】図3の領域708の上位25ビットには4)の場合と同様に親ファイルの次のクラスタの番号が記録され、領域709の下位7ビットにはクラスタ中のフラグメントの使用数の数が記録される。クラスタのフラグメントがすべて使用されていた場合は領域709の下位7ビットには0の値が記録される。領域709の下位7ビットの値が0の場合は、4)の場合と同じ形式のため4)の場合と5)の場合を識別できないが、ディレクトリにてそのファイルが階層ファイルであるか否かを判断できるようになっている。また、フラグメントFATのエントリを調べることでクラスタとして使用されているか否かが分かるため階層ファイルであるか否かを判断できる。

【0057】図4にフラグメントFATのエントリ形式を示す。フラグメントFATのエントリには対応するフラグメントの使用状態に従って、下記の3つの形式で、32ビットの値が記録される。

- 1) 未使用のフラグメント
- 2) クラスタ単位で使用中のフラグメント
- 3) ファイルに割り当て済みのフラグメント

【0058】1)の場合は、そのフラグメントを含むクラスタがフラグメント単位で使用されている、つまり、図3のクラスタFATのエントリ形式の2)、3)、5)の場合で、且つ、そのフラグメントがファイルに割り当てられていない未使用のフラグメントであることを示す。領域801の32ビットには識別子0が記録される。

【0059】2)の場合は、そのフラグメントを含むクラスタがフラグメント単位で使用されていなく、クラスタを単位として使用されている場合、つまり、図3のクラスタFATエントリ形式の1)、4)の場合で、そのクラスタはクラスタFATにより管理されていることを示す。

【0060】領域802の上位25ビットには識別子1が記録され、領域803の下位7ビットには識別子0が記録される。そのクラスタに属するすべてのフラグメントのFATエントリは図4の2)の形式を取る。ボリュームの初期化時にはすべてのクラスタはクラスタ単位で使用されており、すべてのフラグメントFATのエントリはクラスタ単位で使用中のフラグメントとして初期化される。この初期化の件は請求項8を具現化した一例を示す事例である。

【0061】3)の場合は、1)の場合と同様にクラス



タはフラグメント単位で使用されており、且つ、そのフラグメントはファイルに割り当てられている場合を示す。フラグメントFATのエントリにはファイルを構成する次のフラグメントの位置が記録され、領域804の上位25ビットには次のフラグメントが属するクラスタの番号、下位7ビットにはそのクラスタ内でのフラグメントを示すフラグメント番号が記録される。

【0062】次に、上述のクラスタFAT及びフラグメントFATのエントリ形式を使用して本発明の階層管理ファイル装置510の動作の事例を図5を参照して説明する。FATエントリの値は32ビットであり、説明の便宜のため、上位25ビットと下位7ビットをカンマで区切って表現し、以下、数値の後ろにhが付くことにより16進数で表す。

【0063】図5(a)のディレクトリには領域901のFILE1.DAT、領域905のFILE2.DAT、領域909のFILE3.DAT、領域913のFILE4.DAT、領域917のFILE5.DAT、領域921のFILE6.DAT、領域925のFILE7.DAT、領域929のFILE8.DATが記録されている。

【0064】このディレクトリはファイルと同様、ボリューム内のクラスタ、または、フラグメントに記録されている。また、ディレクトリ内には階層ディレクトリの下位のディレクトリ、サブディレクトリが記録され、システム領域から示されたルートディレクトリより階層ディレクトリをたどることにより目的のファイルのディレクトリエントリを検索することができる。

【0065】ディレクトリエントリは各々のファイルに対して、ファイル情報の属性を示すファイル名(領域901、905、909、913、917、921、925、929)、アロケーション方法(領域902、906、910、914、918、922、926、930)、最初の階層の分割領域であるファイルの最初の位置(領域903、907、911、915、919、923、927、931)、ファイルサイズ(領域904、908、912、916、920、924、928、932)のフィールドなどをもつ。ファイル名のフィールドには、例えば、256文字分(256バイト)が割り当てられる。上述したファイル名及びファイルサイズは請求項1に記載したファイルの属性を具現化した事例であり、ファイルの最初の位置は同具現化した最初の階層の分割領域の位置の事例である。また、前述したように図5(a)のディレクトリエントリテーブルは請求項1に記載の第2の管理情報を具現化した事例である。

【0066】アロケーションの方法のフィールドには4バイトが割り当てられる。値が00000000hの場合はファイルはクラスタ単位として割り当てられていることを示し、値が00000001hの場合はファイル

ルはフラグメントを単位として割り当てられていることを示す。また、値がFFFFFFFFhの場合は階層ファイルの親ファイルであることを示し、上記の値以外の場合は階層ファイルの子ファイルであることを示し、親ファイルの開始クラスタの番号が記録される。上述の00000000hや00000001hによりクラスタ単位かフラグメント単位かを示すことは請求項1に記載の階層の種類の情報を見現化した事例である。また、上述のFFFFFFFFhは同記載のファイルの種類の情報を見現化した事例である。

【0067】ファイルの最初の位置のフィールドには4バイトが割り当てられる。ファイルがクラスタ単位で割り当てられている場合は図3の4)の形式、ファイルがフラグメント単位で割り当てられている場合は、図4の3)の場合の形式で、ファイルの最初の位置を示す。ファイルサイズのフィールドには8バイトの値が割り振られ、ファイルサイズが記録される。

【0068】図5(b)はクラスタFATの例であり、領域933~領域964に、それぞれのクラスタに対応したFATエントリが記録される。領域933、領域934で示した領域のFATエントリは予約されており、FATであることを示す識別子、バージョンなどが記録される。すなわち、クラスタFATのエントリはクラスタ番号2から始まる。上述のクラスタFAT及び後述のフラグメントFATは、請求項1に記載の第1の管理情報を具現化した事例である。

【0069】図5(b)の例では、図1において、2次記録装置503のボリュームの容量を1ギガバイト、クラスタサイズを64キロバイトにすると、2次記録装置503は、16、384個のクラスタに分割される。各クラスタエントリは32ビット(4バイト)の記録領域を持ち、領域935のクラスタ番号2のFATエントリより昇順に配置され(領域935、936、937、938、939、940、941、の順、クラスタ番号はこの順に対応して、2、3、4、5、6、7、8となる)、クラスタFATは約64キロバイトのサイズとなる。クラスタFATもファイルと同様にボリュームのクラスタ、またはフラグメントに記録され、その位置はシステム領域から示される。

【0070】図5(c)はフラグメントFATの例であり、フラグメントFATのエントリは対応するクラスタ毎にまとまって配置され、すべてのクラスタに対するフラグメントFATのエントリで構成されている。図5(c)の例では、フラグメントのサイズを512バイトにすると、クラスタは128個のフラグメントに分割される。各エントリは、クラスタ2、フラグメント番号0のフラグメントに対するエントリ965からクラスタFATと同様な昇順に配置され、最初の128個のエントリはクラスタ2に属する。以下、昇順に各クラスタに属するフラグメントFATエントリが配置され、16、3

84個のクラスタに対するフラグメントエントリが配置される。各エントリは32ビットであり、フラグメントFATは合計8メガバイトのサイズになる。フラグメントFATもファイルと同様、ボリューム内のクラスタ、または、フラグメントに記録され、その位置はシステム領域から示される。フラグメントFAT及びクラスタFATのエントリが昇順番号を付されている件は請求項2に記載した件を具現化した事例である。

【0071】図5(a)の領域901のFILE1. DAT、領域905のFILE2. DAT、領域909のFILE3. DATはクラスタ単位でファイルに割り当てられた例である。アロケーション方法のフィールドである領域902、領域906、領域910の値が00000000hになっていることでクラスタ単位でアロケーションされていることが分かる。

【0072】領域901のFILE1. DATのディレクトリエントリの領域903にはファイルの最初の位置、すなわち、ファイルを構成する最初のクラスタの場所が記録されている。領域903で示した上位25ビットの値0000002hより最初のクラスタはクラスタ番号2であることが分かる。

【0073】次に、クラスタFATのクラスタ番号2の領域935のFATエントリの上位25ビットの値0000004hより、次のクラスタのクラスタ番号は4であることが分かる。また、クラスタ番号4の領域937には次のクラスタのクラスタ番号5が記録されており、クラスタ番号5の領域938には、次のクラスタのクラスタ番号8が記録されており、クラスタ番号8の領域941には次のクラスタのクラスタ番号9が記録されているというようにして、領域937、938、941、942、943、944、945、946、947、948、949、950、953、954のクラスタFATのエントリによりFATチェーンをたどり、最後に領域955で示したエントリの上位25ビットの値が1FFFFFFFhであることから、このクラスタがファイルを構成する最後のクラスタであることが分かる。1FFFFFFFhは請求項7に記載したファイルの終了を示す情報を具現化した事例である。

【0074】上記のことより、FILE1. DATはクラスタ2、4、5、8、9、10、11、12、13、14、15、16、17、20、21、22の合計16個のクラスタより構成されていることが分かる。ディレクトリエントリの領域904のファイルサイズの値が1、032、091バイトであり、最後のクラスタ22は49、051バイトのみ使用されていることが分かる。クラスタ単位で構成されたファイルのクラスタFATエントリはすべて図3(4)の場合の形式をとる。

【0075】領域905のFILE2. DAT、領域909のFILE3. DATの場合も同様の方式によってクラスタFATをたどることができ、FILE2. DA

Tはクラスタ7のみで構成され、FILE3. DATはクラスタ23、クラスタ6によって構成されていることが分かる。

【0076】領域913のFILE4. DAT、領域917のFILE5. DATはフラグメント単位でファイルに割り当てられた例である。アロケーション方法のフィールドの領域914及び領域918の値が00000001hであることからフラグメント単位でアロケーションされていることが分かる。

【0077】領域913のFILE4. DATのディレクトリエントリの領域915にはファイルを構成する最初のフラグメントの場所が記録されている。領域915の上位25ビットの値0000012h、下位7ビット00hより、最初のフラグメントはクラスタ番号18：フラグメント番号0であることが分かる。

【0078】図5(c)の最初のフラグメントのフラグメントFATのエントリのための領域969は、クラスタ番号18：フラグメント番号0であり、フラグメントFATの2176番目(クラスタの数(18-2)×フラグメント数128+1)のFATエントリである。その値の上位25ビットの値が0000012h、下位7ビットが01hであり、次のフラグメントはクラスタ番号18、フラグメント番号1であることが分かる。

【0079】次のフラグメントのFATエントリのための領域970、すなわち、クラスタ番号18：フラグメント番号1は上位25ビットが0000012h、下位7ビットが02hで、次のフラグメントはクラスタ番号18：フラグメント番号2であることが分かる。このように、下位階層の分割領域のフラグメントが自分の階層の連結情報及び使用情報と上位階層の分割領域であるクラスタとの連結情報及び使用情報を有することは請求項6を具現化した事例である。

【0080】以下、クラスタ番号18、フラグメント番号2のFATエントリ971、クラスタ番号18：フラグメント番号127のFATエントリ976のFATチェーンをたどり、領域976のFATエントリの上位25ビットが1FFFFFFFhであることからこのフラグメントがファイルを構成する最後のフラグメントであることが分かる。

【0081】上記の操作よりFILE4. DATは、クラスタ番号18：フラグメント番号0、クラスタ番号18：フラグメント番号1、クラスタ番号18：フラグメント番号2、クラスタ番号18：フラグメント番号127の4つのフラグメントより構成されており、領域916のファイルサイズの値が1581であることから領域976の最後のフラグメントは1581バイト-512バイト×3=45バイト使用されていることが分かる。

【0082】領域917のFILE5. DATの場合も同様の方法でフラグメントFATのチェーンをたどることができ、領域919の最初のフラグメントの値より領

域977、領域978のエントリをたどり、ファイルはクラスタ番号18：フラグメント番号3、クラスタ番号19：フラグメント番号1の2つのフラグメントで構成されていることが分かる。

【0083】上記のようにクラスタ18および19はフラグメント単位でファイルのアロケーションに使用されているが、クラスタ18のクラスタFATのエントリ951の上位25ビットは0000000h、下位7ビットは05hであり、このエントリは図3の2)の形式をとり、このクラスタはフラグメント単位で使用されていることを示している。

【0084】また、下位7ビットの値よりクラスタ中の5つのフラグメントが使用中であることを示している。クラスタ19のクラスタFATのエントリ952の上位25ビットは0000000h、下位7ビットは01hであり、クラスタ19中1つのフラグメントを使用中であることを示している。上述したFILE1. DAT～FILE5. DATの読み出しの動作は請求項3を具現化した事例である。

【0085】領域921のFILE6. DATのファイルのアロケーション方法のフィールド922の値がFFFFFFFFhであることからFILE6. DATは階層ファイルの親ファイルであることが分かり、また、領域925のFILE7. DATの領域927のファイルの最初の位置の値と、領域926のアロケーション方法の値と、領域929のFILE8. DATの領域930のアロケーション方法の値とが00000018hであり、領域921のFILE6. DATの領域923のファイルの最初の位置の値00000018hと一致することから領域925のFILE7. DAT、領域929のFILE8. DATは領域921のFILE6. DATの子ファイルであることが分かる。

【0086】階層ファイルの親ファイルのFATチェーンをたどる方法は、通常のクラスタ単位のファイル割り当てのファイルの場合と同じである。領域921のFILE6. DATの領域923のファイルの最初の位置の値00000018hより、最初のクラスタの番号は24であることが分かる。

【0087】次にクラスタ24のクラスタFATのエントリ957の上位25ビット0000019hより次のクラスタの番号25であることが分かる。さらに、クラスタ25のクラスタFATのエントリ958の上位25ビットの値000002Ahより次のクラスタは番号26であることが分かり、最後にクラスタ番号26のクラスタFATのエントリ959の上位25ビットの値がFFFFFFFFhであることよりこのクラスタがファイルを構成する最後のクラスタであることが分かる。

【0088】上記の操作により領域921のFILE6. DATはクラスタ24、25、26の3つのクラスタより構成されていることが分かる。階層ファイルの親

ファイルではすべてのクラスタのデータを使用したと見なされ、領域924のファイルサイズは196、608バイトになる。また、クラスタFATのエントリ957、958、959の下位の7ビットの値03h、03h、02hより、それぞれのクラスタ内のフラグメントはそれぞれ、3個、3個、2個使用されていることが分かる。このように、上位階層であるクラスタが、自分の連結情報及び使用情報を示す情報だけでなく、下位階層であるフラグメントの使用状況を識別できることは請求項5を具現化した事例である。

【0089】領域925のFILE7. DAT、領域929のFILE8. DATは階層ファイルの子ファイルであり、親ファイルの領域921のFILE6. DATに割り当てられたクラスタをさらにフラグメント単位で割り当てが行われる。階層ファイルの子ファイルの場合も、フラグメントFATのチェーンのたどる方法は、上記の通常のフラグメント単位で割り当てられたファイルの場合と同じ方法である。

【0090】領域925のFILE7. DATの場合、領域927の最初のクラスタ番号24：フラグメント番号0からフラグメントFATのエントリ981、983、991をたどることにより、ファイルはクラスタ番号24：フラグメント番号0、クラスタ番号24：フラグメント番号2、クラスタ番号26：フラグメント番号2の3つのフラグメントから構成されていることが分かる。

【0091】領域929のFILE8. DATの場合も、同様に、領域931の最初のクラスタ番号24：フラグメント番号1からフラグメントFATのエントリ982、985、986、987、990をたどることにより、ファイルはクラスタ番号24：フラグメント番号1、クラスタ番号25：フラグメント番号0、クラスタ番号25：フラグメント番号1、クラスタ番号25：フラグメント番号2、クラスタ番号26：フラグメント番号1の5つのフラグメントから構成されていることが分かる。上述したFILE6. DAT～FILE8. DATの読み出しの動作は請求項4を具現化した事例である。

【0092】次に、以上説明した本発明の階層管理ファイル装置の動作の詳細を下記(1)～(14)の順で、図6～32のフローチャートを参照して説明する。これらの動作は全て請求項1に記載のソフトウェア制御手段によりなされる。

- (1) ファイルの読み込み
- (2) クラスタ単位での読み込み
- (3) フラグメント単位での読み込み
- (4) ファイルの書き込み
- (5) クラスタ単位での書き込み
- (6) 新規クラスタの取得
- (7) 検索範囲から新規クラスタを取得

- (8) フラグメント単位の書き込み
- (9) 新規フラグメントの取得
- (10) 検索範囲より新規フラグメントを取得
- (11) クラスタから新規フラグメントを取得
- (12) 階層ファイルの新規フラグメントを取得
- (13) 新規ファイルの作成
- (14) 新規階層ファイルの子ファイルの作成

#### 【0093】(1) ファイルの読みこみの説明

図6及び図7に“ファイルの読みこみ”のフローチャートを示す。“ファイルの読みこみ”はS1000より開始される。ファイルのパス名: PathName (フローチャートに示すPathNameというパラメータがファイルのパス名であることを示す。以下同様の表記を行う)、ファイル内での読み込み開始位置のバイトオフセット: ByteOffset、読み込むバイト数: ByteLength を与える。バイトオフセットはファイルの開始位置から読み込み開始位置までのバイト数であり、読み込み開始位置は各クラスタの開始位置と一致しているとは限らない。

【0094】S1001にてPathNameから階層ディレクトリをルートディレクトリよりたどり、そのファイルのディレクトリエントリを検索する。S1002にてファイルが存在しているか検査し、検索できた場合はS1003に進み、できなかった場合はS1004にてエラーの処理を行う。

【0095】S1003にてファイルのディレクトリエントリよりアロケーション方法: AllocType、ファイルの最初の位置 (クラスタまたはフラグメントを示す): Location、ファイルサイズ: FileSizeを読み取りメモリに保存する。

【0096】S1005にてByteOffset、ByteLengthから、読みこむ開始位置および終了位置がファイルサイズを超えていないか検査する。ファイルサイズ未満である場合はS1006に進み、ファイルサイズ以上の場合はS1112にてエラー処理を行う。

【0097】S1006にてアロケーションの方法を検査し、ファイルがクラスタ単位でアロケートされている場合 (AllocType == 0)、または、階層ファイルの親ファイル (AllocType == FFFFFFFFh) の場合は、S1007にて後述する(2)の“クラスタ単位の読みこみ”をおこなう。

【0098】それ以外の場合、つまり、ファイルがフラグメント単位でアロケートされている場合、階層ファイルの子ファイルの場合は、S1008にて後述する

(3)の“フラグメント単位の読みこみ”をおこなう。S1007、S1008の処理が終了したらS1010に進み“ファイルの読みこみ”を終了する。

【0099】(2) “クラスタ単位の読みこみ”の説明  
図8及び図9に“クラスタ単位の読みこみ”のフローチャートを示す。クラスタ単位の読みこみの場合はクラスタFATのFATエントリのチェーンをたどることにより

読みこみが行われる。クラスタ単位の読みこみはS1101より開始される。ディレクトリエントリより得たファイルの最初の位置: Locationと、ファイル内での読みこみ位置のバイトオフセット: ByteOffsetと、読み込むバイト数: ByteLengthとを与える。

【0100】S1102にて、開始クラスタアドレス: ClusterAddress、ファイル内での読みこみ位置のクラスタオフセット: ClusterStart、ファイル内での読みこみ終了位置のクラスタオフセット: ClusterEndをS1102に示すように演算して求め、クラスタオフセットの変数: ClusterOffset を0に初期化する。クラスタオフセットの変数は整数であり、クラスタが読み込まれるにつれて1個ずつ変化する。Location は図3の4)の形式にて記録されており、Location の上位25ビットの値よりClusterAddressが求まる。クラスタサイズ: ClusterSize はボリューム内で固定の値であり、システムエリアから得ることができる。上述のクラスタアドレスは前述したクラスタ番号に対応するもので、以下も同様である。

【0101】S1103にて、ClusterOffsetが読みこみ開始位置のクラスタオフセット: ClusterStart 以上の場合は読み込むデータが記録されたクラスタであり、S1104に進みクラスタデータの読みこみ処理を行う。ClusterOffsetがClusterStart未満の場合は、S1104の処理をおこなわず、S1105に進む。

【0102】S1104では実際のクラスタデータの読みこみを行う。開始クラスタ: ClusterStartの読みこみの場合でByteOffsetがクラスタサイズの倍数でなかった場合は、開始クラスタ内の読みこみ位置のバイトオフセットはByteOffset%ClusterSizeとなる(%)はByteOffsetとClusterSizeの値のモジュロ演算を示す、以下同様)。また、最終クラスタClusterEndの場合でByteOffset+ByteLength がクラスタサイズの倍数でなかった場合は、最終クラスタ内の読みこみバイト数は(ByteOffset+ByteLength)%ClusterSize となる。それ以外の場合はクラスタ内のすべてのデータが読み込まれる。

【0103】S1105にて現在のクラスタオフセット: ClusterOffset が終了位置のクラスタオフセット: ClusterEndと一致するか検査し、最終クラスタである場合はS1106に進みクラスタ単位の読みこみを終了する。終了クラスタに達していない場合は、S1107に進み現在のクラスタオフセットの値: ClusterOffset を一つ増加させる。次にS1108に進み、クラスタFATから現在のクラスタのFATエントリの値: Entryを読み込む。

【0104】次にS1109にてEntry がファイルを構成する最終クラスタであるか検査する。本来であれば図6のS1005にてファイルサイズの検査をおこなっているため最終クラスタにはなりえないが、図3の4)、5)の形式より上位25ビットの値が1FFFFFFFh

であった場合は、S1110に進みエラー処理を行う。

【0105】最終クラスタでない場合はS1111に進み、次のクラスタアドレスの上位25ビット: Entry.High25Bit を現在のクラスタアドレス: ClusterAddressに代入する。次に、再度、S1103に戻り、S1105にて最終クラスタになるまで、上記操作によって要求したすべてのデータの読みこみが行われる。

【0106】(3)“フラグメント単位の読みこみ”の説明

図10及び図11に“フラグメント単位の読みこみ”のフローチャートを示す。フラグメント単位の読みこみの場合はフラグメントFATのFATエントリのチェーンをたどることにより読みこみが行われる。図8及び図9に示した“クラスタ単位の読みこみ”とまったく同じ手順にて読みこみが行われる。S1208でフラグメントFATが使用されることと、S1202の開始フラグメントアドレス (FragmentAddress)、および、S1209、S1211のEntryが、図4の3)の形式を取る。フラグメントアドレスは前述のフラグメント番号に対応するものであり、以下も同様とする。

【0107】(4)“ファイルの書きこみ”の説明

図12及び図13に“ファイルの書きこみ”のフローチャートを示す。ファイルの書きこみの場合は、図6及び図7に示した“ファイルの読みこみ”の場合とほぼ同じ手順にて行われ、後述する“クラスタ単位の書きこみ”の場合、S1310と、後述する“フラグメント単位の書きこみ”の場合、S1311とに分けて処理される。また、ファイルのサイズの検査はおこなわれず、ファイルの書きこみによって新たにクラスタまたはフラグメントが割り当てられてファイルサイズ (FileSize) が大きくなっていた場合はS1312にてディレクトリエントリの更新が行われる。第2の管理情報であるディレクトリエントリを更新することは、請求項11を具現化した事例である。

【0108】また、システム構成によってはファイルの書きこみ時にディレクトリエントリにファイルが存在しない場合は、新規にファイルを作成することも可能であり、その場合はS1303にて検索ができなかった場合は、S1306にて“新規ファイルの作成”が行われる。

【0109】(5)“クラスタ単位の書きこみ”の説明

図14及び図15に“クラスタ単位の書きこみ”のフローチャートを示す。基本的には“クラスタ単位の書きこみ”は、図8及び図9の“クラスタ単位の読みこみ”の場合と同じ手順であり、S1404にてクラスタへのデータの書きこみをおこなう。読みこみの場合と同様、開始クラスタではクラスタ内でのオフセットがあり、終了クラスタではクラスタデータの端数がおこるが、書きこみではクラスタ内の書きこみを行わないデータを変えないように、クラスタ内の必要なデータだけ更新する必要がある。

る。

【0110】また、書きこみでは書きこむデータがファイルサイズを超える場合は、新規にクラスタをアロケートする必要があり、S1409にて現在のクラスタがファイルを構成するクラスタの最終であった場合は、S1410にて“新規クラスタの取得”がおこなわれる。

【0111】フローチャートの例では必要なたびに新規クラスタの取得を行っているが、S1402にてすべてのデータを書きこむのに必要とするクラスタ数を知ることができ、書きこみを開始するまえにあらかじめ、必要とする新規のクラスタをアロケートしてから書きこみを開始することもできる。その場合は書きこむデータを連続したクラスタに配置されるように、後述する“新規クラスタの取得”にて未使用のクラスタを検索する場合、最初に必要とする個数分の連続した未使用クラスタを探すことによって可能である。このことは、請求項9に記載した件を具現化した事例である。

【0112】(6)“新規クラスタの取得”の説明

図16に“新規クラスタの取得”のフローチャートを示す。新規クラスタの取得では未使用のクラスタを如何にして検索するかが重要になる。図16ではS1502とS1504の2回に検索範囲を分けて新規クラスタの取得を行っている。すなわち、まず、最初にS1502にて現在のクラスタからボリューム内の最後のクラスタまで検索する。つぎに、S1504にてボリュームの先頭のクラスタから現在のクラスタまで検索する。上記の操作にて新規クラスタを取得できなかった場合はS1506にてエラーとなる。

【0113】新規クラスタの検索の基本的な方針としては、現在のクラスタの位置よりできるだけ離れていない位置に未使用クラスタが見つかった方がシークタイムを少なくすることができる。そのためS1502、S1504の範囲をさらに細かく分割して、後ろの範囲、前の範囲という様に順次、現在のクラスタから離れていく方向に検索をおこなうほうが望ましい。

【0114】また、(5)で説明したように、あらかじめ必要とするクラスタの数がわかっている場合は、一つ一つ検索するよりは、必要な個数のクラスタの読み出しのためのシークタイムの和が最小になるような配置場所を探しだすことが望ましい。これを行うためにはクラスタFATの使用状況をあらかじめ検査して、分割した範囲に大体どのくらいの未使用のクラスタがあるかの情報をもっていることにより行うことができる。このことは、請求項1及び請求項5に記載の件を具現化した事例である。

【0115】クラスタの数が多くなると検索の処理に時間がかかるが、本発明の階層管理ファイル装置510ではFATを階層化してあり、クラスタサイズを大きくすることが可能であり、クラスタの数を少なくできるので、より効率的に未使用クラスタの検索が可能である。

【0116】(7) “検索範囲から新規クラスタを取得”の説明

図17及び図18に“検索範囲から新規クラスタを取得”のフローチャートを示す。検索はS1601より開始する。現在のクラスタアドレス: ClusterAddressを、検索範囲の開始のクラスタアドレス: FindCluster に、ファイルの最終クラスタのアドレスを検索範囲の終了のクラスタアドレス (FindEnd) に与える。

【0117】まず、S1602にて検索範囲が検査する。FindCluster は以降、整数変数により変えて、検索中のクラスタアドレスとして使用され、最終クラスタ: FindEnd 以下であるかによって、FindCluster が範囲内であるか検査する。検索範囲であればS1603に進み、検索範囲を超えていた場合はS1611に進み、この場合は検索できなかった形で検索が終了される。

【0118】S1603では、クラスタFATよりクラスタアドレス: FindCluster のエントリ値: Entryを読み込む。次に、S1604にて未使用クラスタであるか確認する。未使用クラスタの場合は図3の1)の形式をとり、値が00000000hに等しい場合は未使用と判断し、S1606に進む。それ以外の場合は、S1605に進み、現在の検索クラスタアドレスを一つ増加させ、S1602に戻り次のクラスタを調べる。この操作は検索範囲の中で未使用クラスタが検索されるまで繰り返される。

【0119】S1604にて未使用クラスタが検索された場合は、S1606にて新しいFATエントリの値を設定する。常にファイルの最終クラスタを示すFATエントリとなり、図3の4)の形式で、上位25ビットのクラスタ番号が1FFFFFFFhの値になる。S1607では、上記FATエントリの値: Entryを検索された未使用のクラスタ: FindClusterのFATエントリ値として書き込み更新する。上述の未使用のクラスタが検索された後、第1の管理情報であるクラスタFATエントリを更新すること、及び後述する未使用フラグメントが検索された後、フラグメントFATエントリを更新することは請求項10に記載の件を具現化した事例である。

【0120】次に、S1608では現在の最終クラスタのFATエントリを検索された未使用クラスタを示すように設定する。図3の4)の形式にて上位25ビットの値としてFindClusterを設定する。S1609にてクラスタアドレス: ClusterAddressのFATエントリに上記Entry値を書き込み更新する。ただし、新規ファイルの作成の場合、一番最初のクラスタを(6)の“新規クラスタの取得”により行う場合は、現在のFATエントリの値は更新せず、ディレクトリエントリで最初の位置を示すようになるので、S1608、S1609の操作は行う必要がない。

【0121】S1606からS1609までの操作によりFATチェーンは更新され、検索されたクラスタ: F

indClusterが現在の最終クラスタになるように、S1610で現在のクラスタアドレス: ClusterAddressに代入し、S1611にて“検索範囲から新規クラスタの取得”が終了する。

【0122】(8) “フラグメント単位の書き込み”

図19及び図20に“フラグメント単位の書き込み”のフローチャートを示す。フラグメント単位の書き込みの手順は、ほぼ、図14及び図15の“クラスタ単位の書き込み”と同じであり、クラスタ単位の場合はディレクトリエントリから示されたファイルの最初の位置(クラスタ)からクラスタFATを使用するのに対し、“フラグメント単位の書き込み”の場合は、フラグメントFATを使用する。書き込みによりファイルサイズを増加する場合は、S1710にて後述する“新規フラグメントの取得”を行う。

【0123】(9) “新規フラグメントの取得”の説明

図21に“新規フラグメントの取得”のフローチャートを示す。“新規フラグメントの取得”の場合も、図16の“新規クラスタの取得”と同様に、検索範囲を分けて新規フラグメントの場所を探す。新規フラグメントの取得はS1802にて階層ファイルの子ファイルの場合を判断し、その場合はS1804にて後述する“階層ファイルの新規フラグメントの取得”をおこなう。

【0124】通常のファイルの場合はS1803に進み、“新規フラグメントの取得”をおこなう。“新規フラグメントの取得”では、まずクラスタFATを参照して、各クラスタ内のフラグメントの使用状況を調べてから、次にフラグメントFATより対応するクラスタの範囲で未使用のフラグメントを探すので、後述する“検索範囲より新規フラグメントの取得”では、現在のフラグメントアドレス: FragmentAddressの上位25ビットの値からクラスタアドレスを求め、まず、S1803にて現在のクラスタからボリュームの最終クラスタの範囲を探し、見つからなかった場合は、S1806にてボリュームの最初のクラスタから現在のクラスタの範囲を探す。新規フラグメントが所得できなかった場合はS1808にてエラー処理を行う。新規フラグメントが取得できた場合は、S1809にて“新規フラグメントの取得”を終了する。

【0125】(10) “検索範囲より新規フラグメントを取得”の説明

図22及び図23に“検索範囲より新規フラグメントの取得”のフローチャートを示す。S1901より“検索範囲より新規フラグメントの取得”は開始され、(9)の“新規フラグメントの取得”の説明で述べたように、新規フラグメントを探す検索範囲として開始クラスタ: FindCluster、検索範囲の終了クラスタ: FindEnd、現在のファイルの最終フラグメントアドレス: FragmentAddressを与える。FindCluster は以降、整数変数により変わり、検索中のクラスタを示す。新規フラグメントの検索

では、最初にクラスタFATを検査し、次に、フラグメントFATを検査するという2段階で行われる。上述のように、上位階層のクラスタから検索を始め、下位階層のフラグメントへと検索を進める件は請求項12の件を具現化した事例である。

【0126】S1902にて検索中のクラスタが検索範囲内か調べる。FindClusterがFindEnd以下であった場合は検索範囲としてS1903に進み、それ以外の場合はS1909に進み、新規フラグメントが取得できなかったとして終了する。検索範囲であった場合は、S1903にて検索中のクラスタ：FindClusterのクラスタFATのエントリ値：ClusterEntryを読み込む。S1904にてエントリ値が図3の1)の形式、つまり、00000000hであった場合は未使用クラスタであり、S1907にすすみ、後述する”B) 未使用クラスタからの新規フラグメントの取得”をおこなう。

【0127】次にS1905にてエントリ値が図3の2)の形式、00000100h未満の場合はそのクラスタに未使用フラグメントがあると判断し、S1908に進んで、後述する”A) 使用中のクラスタからの新規フラグメントの取得”をおこなう。S1904、S1905のいずれでもない場合は、S1906にて検索クラスタアドレスFindClusterを一つ増加させ、S1902に進み、検索範囲内でS1902からS1905の操作を繰り返す。

【0128】S1907およびS1908で新規フラグメントの取得がおこなわれ、S1909にて”検索範囲より新規フラグメントを取得”を終了する。S1907の後述する”A) 未使用クラスタから新規フラグメントを取得”をおこなうと、クラスタがフラグメント単位での使用になるので、もし、ボリューム内の容量を有効に使用したい場合は、S1904の判断を行わず、検索範囲でS1905の判断のみをおこない、それでもなかった場合は再度、S1904の判断にてS1907の操作により未使用クラスタの使用をおこなうなどの方法もとることができる。

【0129】また、S1905にてClusterEntryの下位7ビットの使用中のフラグメントの値より、この後、新規に取得されるフラグメントができるだけ同じクラスタ内に配置されるように、今回の書きこみで必要とする新規フラグメントの数を基に、未使用のフラグメントが多い、クラスタを選択する方法などもとることができる。未使用のフラグメントの多いクラスタを捜すことは、請求項5に記載した件を具現化した事例である。

【0130】(11) ”クラスタから新規フラグメントの取得”の説明

図24、図25及び図26に”クラスタから新規フラグメントを取得”のフローチャートを示す。 ”クラスタからの新規フラグメントの取得”は2つの場合に分けられ、既にそのクラスタがフラグメント単位で使用中の場合

は”A) 使用中のクラスタから新規フラグメントを取得”と、そのクラスタが未使用クラスタである場合は、”B) 未使用クラスタから新規フラグメントを取得”になる。

【0131】”A) 使用中のクラスタから新規フラグメントを取得”の場合はS2001より開始され、現在のフラグメントアドレス：FragmentAddress、検索するクラスタアドレス：FindCluster、検索するクラスタのFATエントリの値：ClusterEntryを与える。S2002にて検索する範囲を対象のクラスタになるように、検索範囲の開始フラグメントアドレス、終了のフラグメントアドレスを設定する。クラスタ内の先頭のフラグメントのアドレスは、上位25ビットにクラスタの値、下位7ビットには0を代入する。クラスタには128個のフラグメントがあり、終了のフラグメントのアドレスは開始フラグメントに127を足した値になる。以降、FindFragmentは整数を変数として、現在検索中のフラグメントを示す。

【0132】S2003にて検索中のフラグメントが範囲内かを検査し、終了クラスタ未満である場合は、S2004に進み、そうでない場合はS2005に進みエラーとなる。本来、クラスタFATのエントリ値より未使用のフラグメントは必ず存在するので、エラーになることはない。

【0133】S2004ではフラグメントFATより検索中のフラグメント：FindFragmentのエントリ値：FragmentEntryを読み込む。S2006にて未使用のフラグメントかを検査し、エントリ値が図4の1)の形式、つまり、00000000hであった場合は、未使用クラスタが検索され、S2008に進む。

【0134】未使用フラグメントでなかった場合はS2007に進み、検索中のフラグメント：FindFragmentの値に1つ増加させ、S2003にすすみ、検索範囲の間、S2003からS2007までの操作を繰り返す。

【0135】未使用フラグメントが発見された場合は、そのフラグメントの存在するクラスタのFATエントリ値：ClusterEntryを1つ増加させ、使用中のフラグメントの値を増やす。次にS2011にすすみ、クラスタアドレス：FindClusterのクラスタFATのエントリの値をClusterEntryに更新する。

【0136】S2012ではフラグメントFATのエントリ値：FragmentEntryがファイルの最終フラグメントを示すように、図4の3)の形式、つまりFFFFFF80hに設定し、S2013にて検索した未使用のフラグメントアドレス：FindFragmentのフラグメントFATのエントリ値：FragmentEntryを書きこみ更新する。

【0137】S2014ではフラグメントFATのエントリ値：FragmentEntryが、検索した未使用のフラグメントアドレス：FindFragmentを示すように設定し、S2015にて現在のファイルのフラグメントアドレス：Fr

agmentAddress のフラグメントFATのエントリ値として更新する。ただし、FragmentAddress が0の場合はディレクトリエントリから示されているので更新は行わない。

【0138】S2012からS2015の操作にて、フラグメントFATのファイルのチェーンは更新され、S2016にて検索した未使用のフラグメントをファイルの最終フラグメントアドレスに設定し、S2017にて終了する。

【0139】“B) 未使用のクラスタから新規フラグメントを取得”はS2009より開始され、未使用のクラスタの場合は、そのクラスタ内のすべてのフラグメントが未使用の状態であるので、最初のフラグメントを新規フラグメントとする。S2010にて未使用のクラスタを使用中にするために、図3の1)の形式に従い下位7ビットに1つフラグメントが使用中であるように1を代入する。また、検索された未使用のフラグメントアドレスとして上位25ビットにクラスタアドレス、下位7ビットは0を代入する。

【0140】以下S2011に進み、“A) 使用中のクラスタから新規フラグメントを取得”の場合と同様に、S2011にてクラスタFATのエントリを更新、S2013にて検索した未使用フラグメントのフラグメントFATのエントリの更新、S2015にてファイルの現在のフラグメントアドレスを更新して、S2017にて終了する。

【0141】(12) “階層ファイルの新規フラグメントの取得”の説明

図27及び図28に“階層ファイルの新規フラグメントを取得”のフローチャートを示す。階層ファイルの子ファイルの場合は、親ファイルに割り当て中のクラスタの中から新規フラグメントを探す。未使用のフラグメントがない場合は、親ファイルに新規クラスタを割り当て、そのクラスタより未使用のフラグメントを検索する。S2101から開始し、現在のフラグメント: FragmentAddress、アロケーション方法: AllocType を与える。上述のことは、請求項13に記載の階層ファイルの新規分割領域の検索及び割り当ての件を具現化した事例である。

【0142】S2102にて、階層ファイルの子ファイルの場合はディレクトリエントリのアロケーション方法: AllocType に親ファイルの開始アドレスが設定されているので、それを最初の検索クラスタのアドレス: ClusterAddressとする。

【0143】S2103で、クラスタFATよりクラスタアドレスClusterAddressのエントリ値: ClusterEntryを読み込む。S2104にて、エントリ値は図4の3)の形式を取り、そのクラスタに未使用のフラグメントがあるかを検査する。下位7ビットの値が0以外の場合は、S2107に進み、前述の“A) 使用中のクラスタ

から新規フラグメントの取得”を行う。

【0144】S2104にて未使用フラグメントがない場合はS2105にて最終クラスタかを検査し、エントリ値が図3の5)の形式にてクラスタ番号が1FFFFFhの場合は、ファイルの最終クラスタと判断し、S2108に進み親ファイルの前述の(6)の“新規クラスタの取得”をおこなう。

【0145】S2105にてクラスタが最終クラスタでない場合は、S2106にてエントリ値の上位25ビットを次のクラスタアドレスとして設定し、S2104にもどり、S2104からS2106の操作を繰り返す。S2107にて前述の“A) 使用中のクラスタから新規フラグメントを取得”が終わったならば、S2111に進み、終了する。

【0146】S2108にて前述の(6)“新規クラスタの取得”をおこなう。新規クラスタの取得にて親ファイルのディレクトリエントリを更新する必要があるため、S2109にて親ファイルの開始アドレス: AllocType を使用して、同じディレクトリに存在する親ファイルを検索して、ファイルサイズ等を更新する。S2110にてそのクラスタから前述の“B) 未使用のクラスタから新規フラグメントを取得”をおこない、S2111にて終了する。

【0147】(13) “新規ファイルの作成”の説明  
図29及び図30に“新規ファイルの作成”のフローチャートを示す。新規ファイルの作成は図12及び図13の“ファイルの書きこみ”の場合にファイルが存在しなかった場合、および、新たにファイルを作成する場合に行われる。ただし、階層ファイルの子ファイルの場合は、親ファイルが存在していた場合のみ作成可能で、このフローチャートは図31及び図32にて示す。

【0148】“新規ファイルの作成”はS2201より開始し、作成するファイルのパス名: PathName、アロケーションの方法: AllocType を指定する。S2202にてパス名: PathNameのファイルのディレクトリエントリを作成し、S2203にてファイルが作成できたか検査し、作成できた場合はS2204に進み、できなかった場合はS2205に進みエラーになる。

【0149】S2204では作成するファイルが階層ファイルの親ファイルかを検査し、AllocTypeがFFFFFFFFhであった場合は、S2211に進み、親ファイルの作成を続ける。次にS2206にてクラスタ単位の作成であるかを検査し、AllocType が0であった場合はS2209に進みクラスタ単位のファイルの作成を続ける。それ以外の場合はS2207に進みフラグメント単位のファイルの作成を続ける。階層ファイルの親ファイルの場合はS2211にて前述した(6)の“新規クラスタの取得”を行う。ClusterAddress を0にすることによりファイルにはまだクラスタがないことを示す。

【0150】次にS2212にてディレクトリエントリ



の値を設定し、AllocType には階層ディレクトリの親であることを示すFFFFFFFFh、ファイルの最初の位置: Locationの上位25ビットにはS2211にて新規に取得したクラスタアドレス: ClusterAddressを設定、ファイルサイズは0にする。

【0151】クラスタ単位のファイルの作成の場合もS2209にて同様に前述した(6)の“新規クラスタの取得”を行い、S2210にてディレクトリエントリを設定する。この場合はAllocTypeにはクラスタ単位であることを示す0を設定する。

【0152】フラグメント単位のファイルの作成の場合は、S2207にて前述した(9)の“新規フラグメントの取得”を行い、S2208にてディレクトリエントリを設定する。この場合はAllocType はフラグメント単位であることを示す1を設定し、ファイルの最初の位置: Location にはS2207にて取得した新規フラグメントのアドレスを設定する。それぞれの場合にディレクトリエントリが設定されたのち、2213にてディレクトリエントリの更新を行い、2214にて終了する。

【0153】(14)“新規に階層ファイルの子ファイルを作成”の説明

図31及び図32に“新規に階層ファイルの子ファイルを作成”のフローチャートを示す。階層ファイルのファイルは親ファイルと同じディレクトリに存在し、S2301では親ファイルのパス名: ParentPathName、子ファイルのファイル名: FileNameを与える。ただし、例では同じディレクトリに作成するが、交互にそのディレクトリを参照できるようにすれば、他のディレクトリなどに作成するなどの方法でもよい。

【0154】まず、最初に、S2302では親ファイルのパス名より親ファイルを検索し、ディレクトリエントリを取得する。S2303にて親ファイルが検索できたかを検査し、できた場合はS2304に進み、できなかった場合はS2305にてエラーとなる。次にS2306にて階層ファイルの親かどうか確認する。ParentAllocType がFFFFFFFFhの場合は、S2307に進み、それ以外の場合はS2308にてエラーとなる。

【0155】次にS2307にて同じディレクトリに子ファイルをファイル名: FileNameにて作成する。S2309にて作成できた場合は2310にすすみ、できなかった場合はS2311にてエラーとなる。

【0156】S2310では現在のファイルのフラグメントアドレス: FragmentAddress に0、親ファイルの最初の位置ParentLocation をAllocTypeに指定して、前述の(12)の“階層ファイルの新規フラグメントの取得”をおこなう。S2312では新規に取得したフラグメントのアドレスFragmentAddress を子ファイルの最初の位置: Locationに設定し、S2313にて子ファイルのディレクトリエントリを更新して、S2314にて終了す

る。

【0157】本実施の形態例では、ファイルを構成するクラスタ、フラグメントをFATチェーンにて表現しているが、ボリュームの割り当てのための記録領域の使用情報をビットマップにより管理し、ファイルの構成情報は別の方式にて管理するような場合でも、クラスタ、フラグメントの管理に各々にビットマップを用意して、クラスタ単位のファイルとフラグメント単位のファイルの割り当て管理が可能であり、同等の効果が実現できる。

10 【0158】また、クラスタビットマップにはそのクラスタの使用中のフラグメントの使用数などの情報も付加し、上位階層の第1の管理情報にて下位階層の第1の管理情報を分かるようにすることができる。このことは、請求項14に記載した件を具現化した事例である。

【0159】また、本実施の形態例では、クラスタ、フラグメントの2階層にて管理を行ったが、さらに、フラグメントをさらに小さな単位に分割して、大きなボリュームに関して多層的に管理情報をもって行うことも可能である。

20 【0160】また、本実施の形態例では、情報記録媒体全体をボリュームとして扱ったが、物理的に別の情報記録媒体を論理的に一つのボリュームとして扱う場合、1つの情報記録媒体の中を複数のボリュームに分割して使用するような場合でも適用可能である。

【0161】本実施の形態例では、すべてのクラスタに対して、フラグメントFATのエントリを持つが、別にクラスタ毎にフラグメントFATのエントリを示す管理テーブルを設け、フラグメントFATのエントリを間接参照することにより、フラグメント単位に利用していないクラスタに対しては、フラグメントFATのエントリを持たず、新規にクラスタをフラグメント単位に利用する場合のみ、新たにそのためのフラグメントFATのエントリをボリューム内に作成するなどの方法を取ることができる。

【0162】これによりフラグメントFATのサイズを小さくすることができ、ボリュームの容量を有効に使用できる。また、必要とするフラグメントFATをまとめることができ、参照のための2次記録装置503へのアクセスを効率良くおこなうことができる。

40 【0163】本実施の形態例では、ファイルの割り当て単位は一つのファイルでは不変であったが、クラスタFATのエントリよりフラグメントFATのエントリの位置を示すことができるように、エントリ値のアドレスのサイズを変更することにより、一つのファイル中で割り当て単位を変えることも可能である。

【0164】本実施の形態例ではクラスタおよびフラグメントは一定のサイズであったが、クラスタFATのエントリ、フラグメントFATのエントリに、そのクラスタまたはフラグメント内での使用部分を記述できるように、使用領域の開始オフセットおよび終了オフセットを

指定するフィールドを設けることにより、クラスタ、フラグメントを可変長の記録ユニットとしてファイルを構成することができ、ファイルのマージ、ファイル中のデータ領域の削除などの編集をファイルのデータをコピーすることなく行えることができる。

【0165】本発明の階層管理ファイル装置510を具備した電子機器として、図1に一例を示したが、本発明の階層管理ファイル装置510はこの例に限らず、連続して、且つ高速に記録再生する必要のあるファイルと、離散的でも構わないファイルの混在するマルチメディア情報

【0166】

【発明の効果】本発明の階層管理ファイル装置及びこれを具備した電子機器によれば、ファイルを複数の割り当てられた階層別の分割領域によって管理でき、映像、音声などのリアルタイム再生、録音を要求する場合は、連続した大きな記録単位長で割り当てられた分割領域にファイルを記録配置することができ、フラグメンテーションによるファイルアクセスの速度の低下を防止することができる。

【0167】また、小さなサイズのファイルの場合は小さな記録単位長で分割領域の割り当てを行うことができるため、記録領域の容量を無駄にすることなく利用できる。

【0168】また、割り当て記録単位長の大きな上位階層の第1の管理情報のデータは、すべて同一の記録単位長で割り当てをおこなった場合よりも小さくでき、第1の管理情報の参照のためのメモリーサイズおよびディスクアクセス、CPUの処理などに要する負荷を低減することができる。

【0169】これにより、大きな記録単位長をファイルに割り当てる場合は、効率よく未使用の割り当て分割領域を探すことができ、さらには、大きな範囲で第1及び第2の管理情報を参照できるためフラグメンテーションがおきないよう最適化も行うことが容易である。

【0170】また、割りあてられた分割領域に階層的に第1の管理情報を持ち、上位階層の第1の管理情報には下位階層に割り当てられた分割領域の使用状況の統計情報を持つことにより、下位階層に割り当てられた分割領域の単位でファイルを記録する場合でも、上位階層の第1の管理情報を参照することにより、すべての下位階層の第1の管理情報を参照する必要がなくなり、小さな記録単位長で割り当てられた分割領域の単位で記録する場合でもフラグメンテーションが起きないように効率的におこなうことができる。

【0171】また、階層ファイルにより、親ファイルが関連する子ファイルの情報を大きな記録単位長で割り当てられた分割領域にまとめることができ、当該階層ファイルを一括してアクセスするような場合は、アクセス速度を速くすることができる。さらに、階層ファイルの機

構を利用して、親ファイルにディレクトリとしての属性を持たせ、ディレクトリとして扱えば、ディレクトリ内のファイルを大きな記録単位長で分割した分割領域内に全てまとめることができる。このことにより、ディレクトリ内のファイルのアクセス速度をさらに速くすることができ、且つディレクトリごとファイルを削除する場合においても、大きな記録単位長の領域を開放し得るため、未使用の記録領域が分散してしまうことを防ぐ効果がある。

【0172】また、階層ファイルの機構を利用することにより、大きな記録単位長で分割された分割領域を連続した小さな記録単位長で分割された分割領域で割り当てする場合の方針を示すことができ、連続した小さな記録単位長で分割された分割領域のまとまりを、新たな割り当て単位として導入することが可能である。

【0173】また、大きな記録単位長で分割された分割領域の場合は参照する第1の管理情報も小さいので、記録、再生する前にファイルの読み出し、書きこみの速度を物理的なファイルの配置位置より容易に計算することができ、アクセス速度の保証できる階層管理ファイル装置が実現できる。

【図面の簡単な説明】

【図1】本発明の実施の形態例の階層管理ファイル装置及びこれを具備した電子機器の概略構成図である。

【図2】本発明の実施の形態例の情報記録媒体の記録領域の構造図である。

【図3】本発明の実施の形態例のクラスタFATエントリ形式を示す図表である。

【図4】本発明の実施の形態例のフラグメントFATエントリ形式を示す図表である。

【図5】図3及び図4のクラスタFATエントリ形式及びフラグメントFATエントリ形式を採用した本発明の階層管理ファイル装置の実施の形態例の動作を説明する説明図である。

【図6】本発明の実施の形態例のソフトウェア制御手段によるファイルの読み込みを示すフローチャートの一部である。

【図7】本発明の実施の形態例のソフトウェア制御手段によるファイルの読み込みを示すフローチャートの他の一部である。

【図8】本発明の実施の形態例のソフトウェア制御手段によるクラスタ単位の読み込みを示すフローチャートの一部である。

【図9】本発明の実施の形態例のソフトウェア制御手段によるクラスタ単位の読み込みを示すフローチャートの他の一部である。

【図10】本発明の実施の形態例のソフトウェア制御手段によるフラグメント単位の読み込みを示すフローチャートの一部である。

【図11】本発明の実施の形態例のソフトウェア制御手

段によるフラグメント単位の読み込みを示すフローチャートの他の一部である。

【図12】本発明の実施の形態例のソフトウェア制御手段によるファイルの書き込みを示すフローチャートの一部である。

【図13】本発明の実施の形態例のソフトウェア制御手段によるファイルの書き込みを示すフローチャートの他の一部である。

【図14】本発明の実施の形態例のソフトウェア制御手段によるクラスタ単位の書き込みを示すフローチャートの一部である。

【図15】本発明の実施の形態例のソフトウェア制御手段によるクラスタ単位の書き込みを示すフローチャートの他の一部である。

【図16】本発明の実施の形態例のソフトウェア制御手段による新規クラスタの取得を示すフローチャートである。

【図17】本発明の実施の形態例のソフトウェア制御手段による検索範囲から新規クラスタ取得を示すフローチャートの一部である。

【図18】本発明の実施の形態例のソフトウェア制御手段による検索範囲から新規クラスタ取得を示すフローチャートの他の一部である。

【図19】本発明の実施の形態例のソフトウェア制御手段によるフラグメント単位の書き込みを示すフローチャートの一部である。

【図20】本発明の実施の形態例のソフトウェア制御手段によるフラグメント単位の書き込みを示すフローチャートの他の一部である。

【図21】本発明の実施の形態例のソフトウェア制御手段による新規フラグメントの取得を示すフローチャートである。

【図22】本発明の実施の形態例のソフトウェア制御手段による検索範囲より新規フラグメントを取得を示すフローチャートの一部である。

【図23】本発明の実施の形態例のソフトウェア制御手段による検索範囲より新規フラグメントを取得を示すフローチャートの他の一部である。

【図24】本発明の実施の形態例のソフトウェア制御手段による使用中のクラスタから新規フラグメントを取得を示すフローチャートの一部である。

【図25】本発明の実施の形態例のソフトウェア制御手段による使用中のクラスタから新規フラグメントを取得を示すフローチャートの他の一部である。

【図26】本発明の実施の形態例のソフトウェア制御手

段による使用中のクラスタから新規フラグメントを取得を示すフローチャートのさらに他の一部である。

【図27】本発明の実施の形態例のソフトウェア制御手段による階層ファイルの新規フラグメントを取得を示すフローチャートの一部である。

【図28】本発明の実施の形態例のソフトウェア制御手段による階層ファイルの新規フラグメントを取得を示すフローチャートの他の一部である。

【図29】本発明の実施の形態例のソフトウェア制御手段による新規ファイルを作成を示すフローチャートの一部である。

【図30】本発明の実施の形態例のソフトウェア制御手段による新規ファイルを作成を示すフローチャートの他の一部である。

【図31】本発明の実施の形態例のソフトウェア制御手段による新規に階層ファイルの子ファイルを作成を示すフローチャートの一部である。

【図32】本発明の実施の形態例のソフトウェア制御手段による新規に階層ファイルの子ファイルを作成を示すフローチャートの他の一部である。

【図33】従来のパーソナルコンピュータ用の磁気ディスクの記録パターンを示したものである。

【図34】図33の磁気ディスクの記録パターンのクラスタの配置を示した配置図である。

【図35】図33の磁気ディスクのディレクトリエントリ配置を示した配置図である。

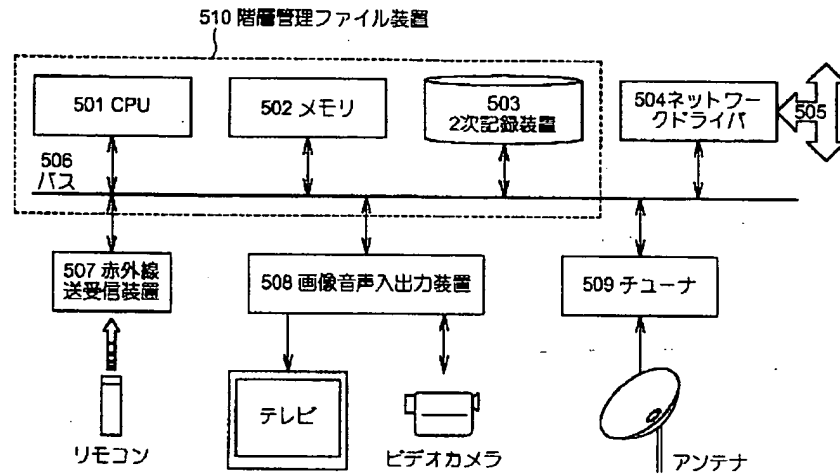
【図36】図33の磁気ディスクのFATのエントリ配置を示した配置図である。

【図37】従来のファイル装置のフラグメンテーションを説明する説明図である。

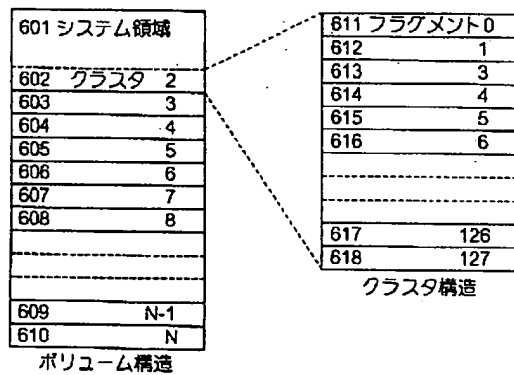
【符号の説明】

501…CPU、502…メモリ、503…2次記録装置、504…ネットワークドライバ、506…バス、507…赤外線送受信装置、508…画像音声入出力装置、509…チューナ、510…階層管理ファイル装置、601…システム領域、602～610…クラスタ、611～618…フラグメント、701～709…クラスタFATエントリ形式、801～805…フラグメントFATエントリ形式、901～932…ディレクトリエントリ、933～964…クラスタFATエントリ、965～996…フラグメントFATエントリ(601, 602～610, 611～618, 701～709, 801～805, 901～932, 933～964, 965～996については領域とも記す)、S1000～S2314…制御ステップ

【図1】



【図2】



【図3】

1.未使用のクラスタ	
32ビット	
701	0 (識別子)

2.フラグメント単位で使用中のクラスタ	
25ビット	7ビット
702 0 (識別子)	703 使用中のフラグメント数 (1から127)

3.フラグメント単位で使用中のクラスタ (すべてのフラグメントを使用中)	
25ビット	7ビット
704 1 (識別子)	705 0 (識別子)

4.ファイルに割り当て済みのクラスタ	
25ビット	7ビット
706 クラスタ番号 (2以上)	707 0 (識別子)

5.ファイルに割り当て済みのクラスタ (階層ファイルにてフラグメント単位で使用中)	
25ビット	7ビット
708 クラスタ番号 (2以上)	709 使用中のフラグメント数 (0から127)

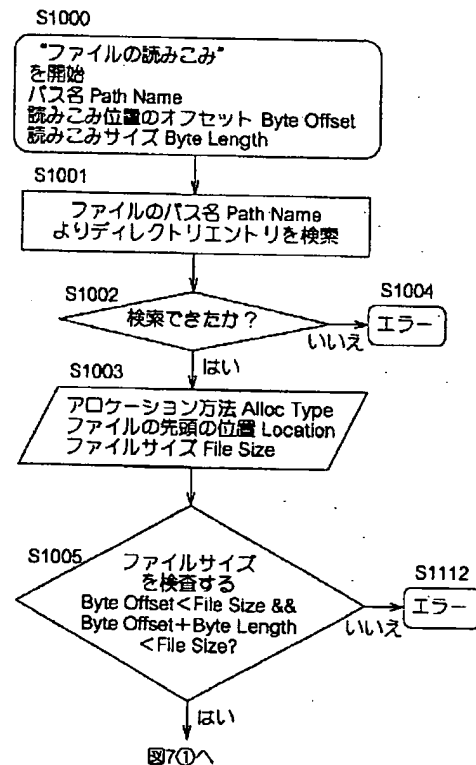
【図4】

1.未使用のフラグメント			
		32ビット	
801	0 (識別子)		
2.クラスタ単位で使用中のフラグメント			
25ビット		7ビット	
802	1 (識別子)	803	0 (識別子)
3.ファイルに割り当て済みのフラグメント			
25ビット		7ビット	
804	クラスタ番号 (2以上)	805	フラグメント番号 (0から127)

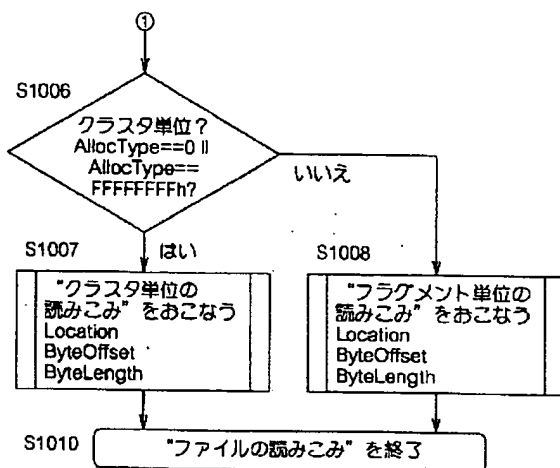
【図5】

ディレクトリ (a)			
901 FILE1.DAT	902 00000000h	903 0000002h,00h	904 1032091
905 FILE2.DAT	906 00000000h	907 0000007h,00h	908 65536
909 FILE3.DAT	910 00000000h	911 0000017h,00h	912 65537
913 FILE4.DAT	914 00000001h	915 0000012h,00h	916 1518
917 FILE5.DAT	918 00000001h	919 0000012h,04h	920 1024
921 FILE6.DAT	922 FFFFFFFFh	923 0000018h,00h	924 196608
925 FILE7.DAT	926 00000018h	927 0000018h,00h	928
929 FILE8.DAT	930 00000018h	931 0000018h,01h	932
クラスタFAT (b)			
933	934	935 0000004h,00h	936 0000000h,00h
937 0000005h,00h	938 0000008h,00h	939 1FFFFFFFh,00h	940 1FFFFFFFh,00h
941 0000009h,00h	942 000000Ah,00h	943 000000Bh,00h	944 000000Ch,00h
945 000000Dh,00h	946 000000Eh,00h	947 000000Fh,00h	948 0000010h,00h
949 0000011h,00h	950 0000014h,00h	951 0000000h,05h	952 0000000h,01h
953 0000015h,00h	954 0000016h,00h	955 1FFFFFFFh,00h	956 0000006h,00h
957 0000019h,03h	958 000002Ah,03h	959 1FFFFFFFh,02h	960 0000000h,00h
961 0000000h,00h	962 0000000h,00h	963 0000000h,00h	964 0000000h,00h
フラグメントFAT (c)			
965 0000001h,00h	966 0000001h,00h	967 0000001h,00h	968 0000001h,00h
969 0000012h,01h	970 0000012h,02h	971 0000012h,FFh	972 0000013h,01h
973 0000000h,00h	974 0000000h,00h	975 0000000h,00h	976 1FFFFFFFh,00h
977 0000000h,00h	978 1FFFFFFFh,00h	979 0000000h,00h	980 0000000h,00h
981 0000018h,02h	982 0000019h,00h	983 000001Ah,02h	984 0000000h,00h
985 0000019h,01h	986 0000019h,02h	987 000001Ah,01h	988 0000000h,00h
989 0000000h,00h	990 1FFFFFFFh,00h	991 1FFFFFFFh,00h	992 0000000h,00h
993 0000001h,00h	994 0000001h,00h	995 0000001h,00h	996 0000001h,00h

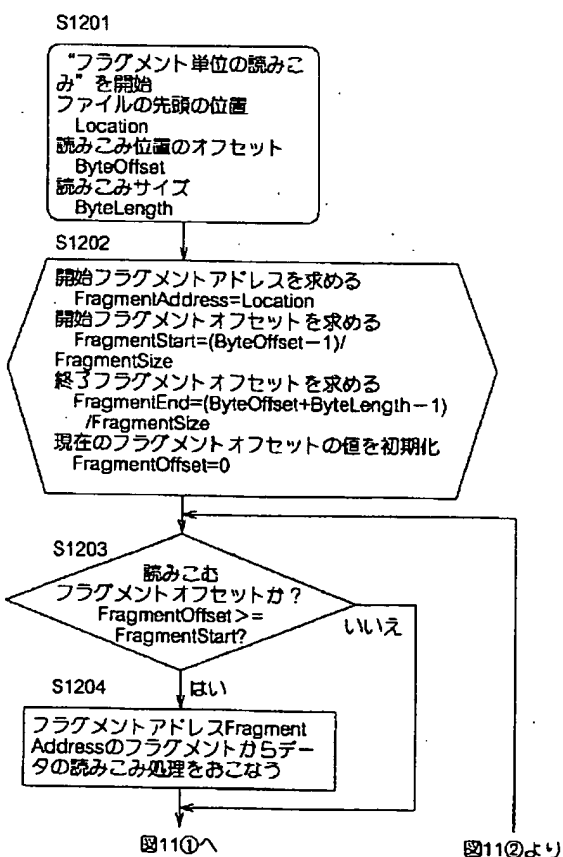
【図6】



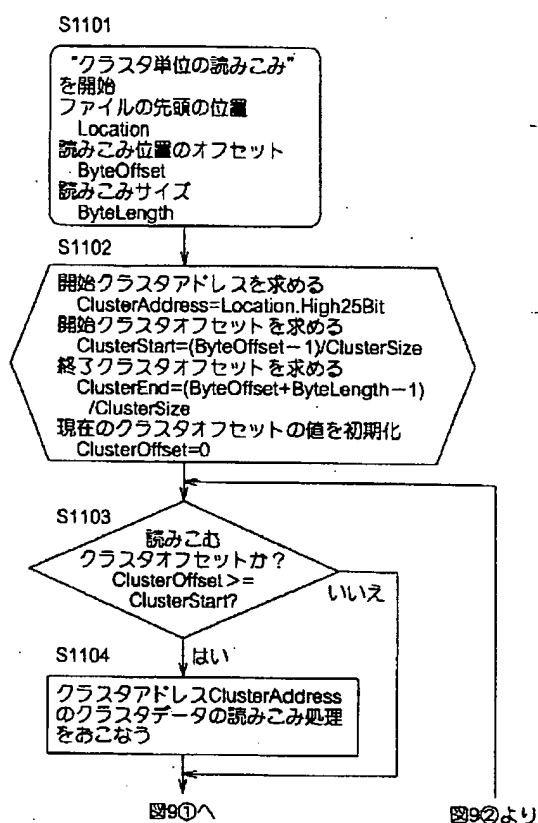
【図7】



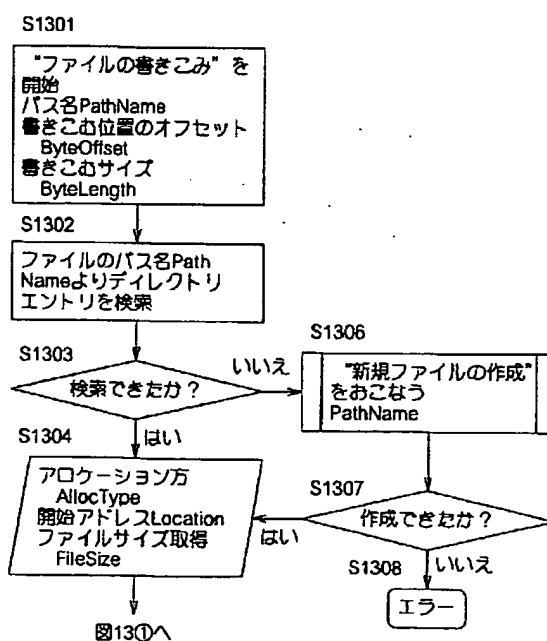
【図10】



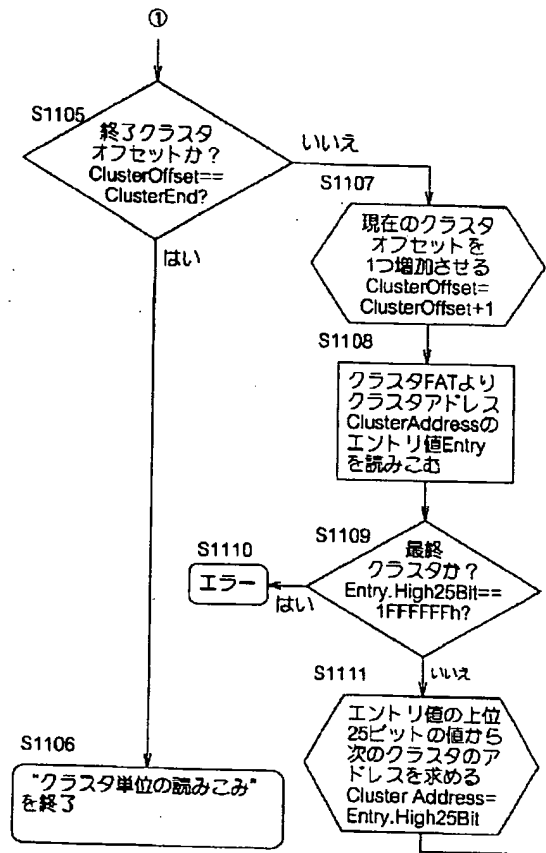
【図8】



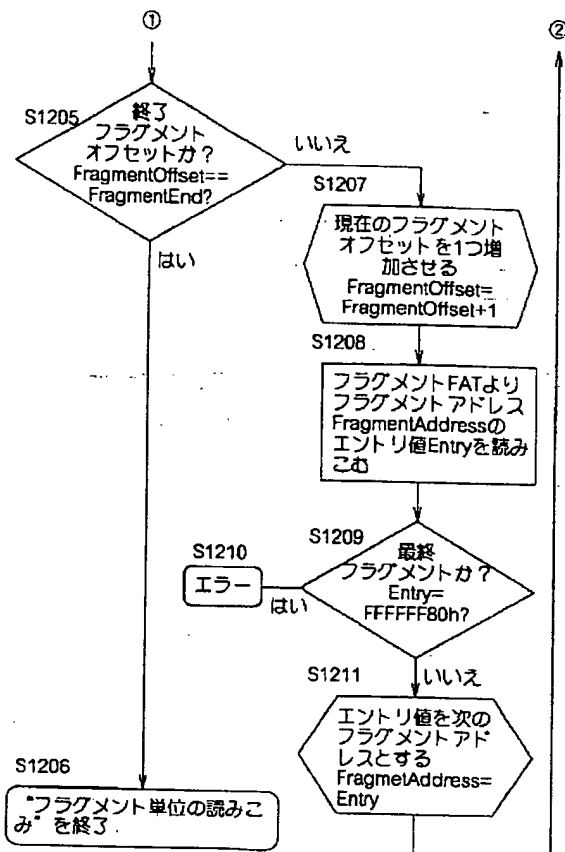
【図12】



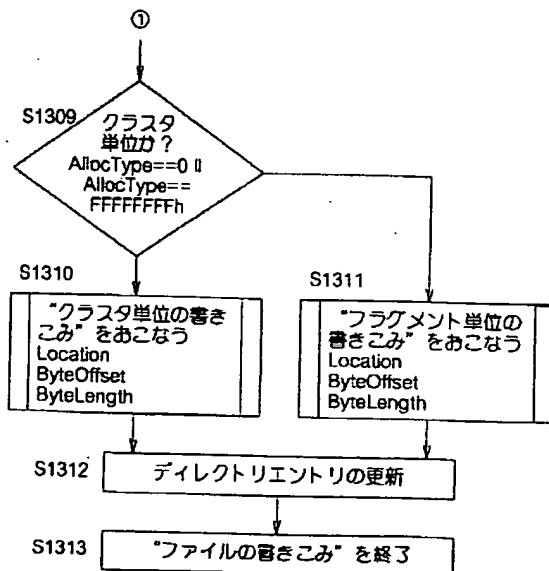
【図9】



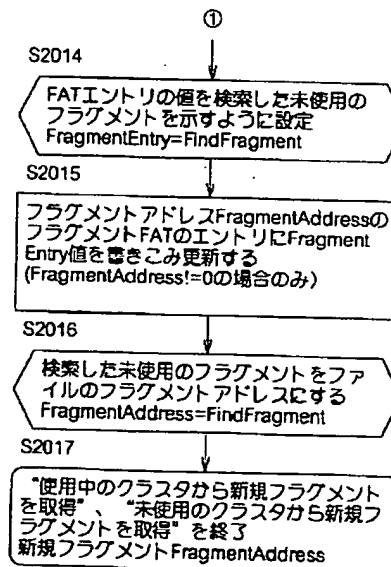
【図11】



【図13】



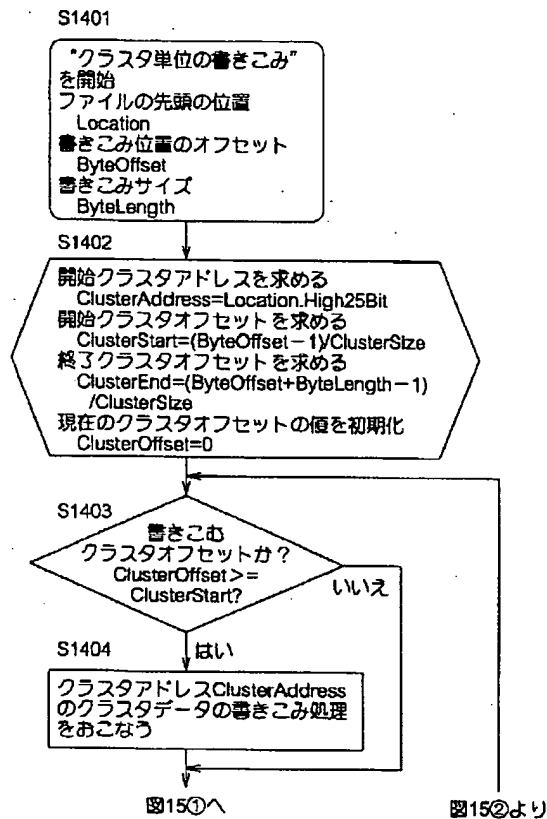
【図26】



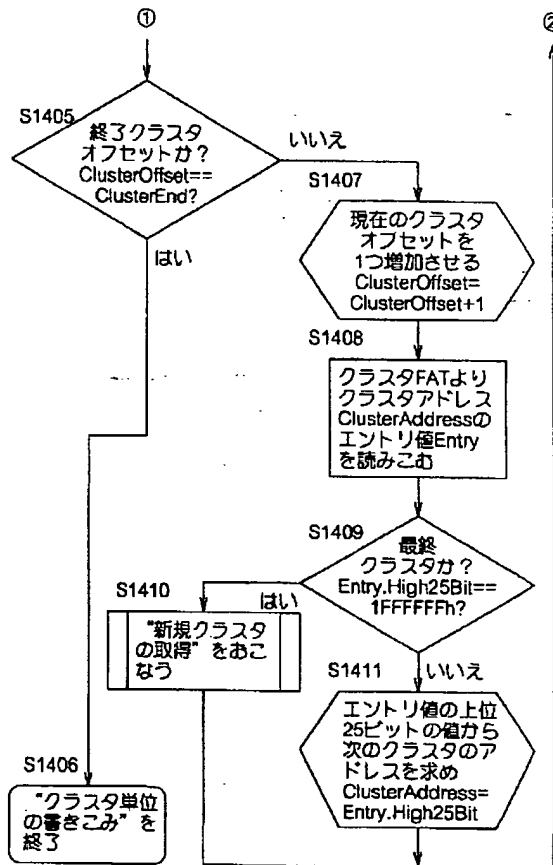
【図34】

101	システム領域
102	クラスタ 2
103	3
104	4
105	5
106	6
107	7
108	8
109	9
110	10
111	11
112	12
113	N-1
114	N

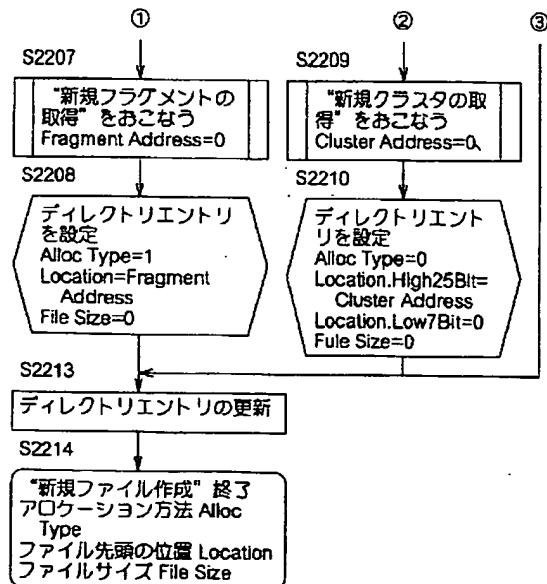
【図14】



【図15】



【図30】

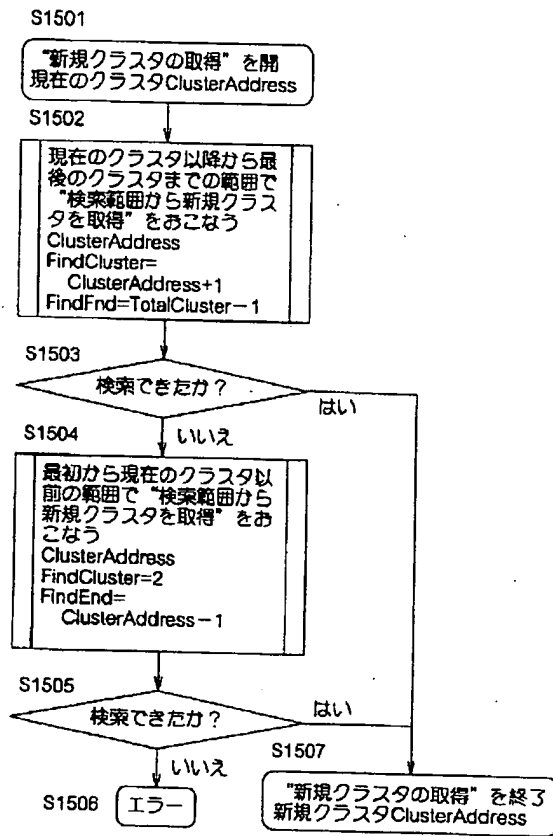


【図35】

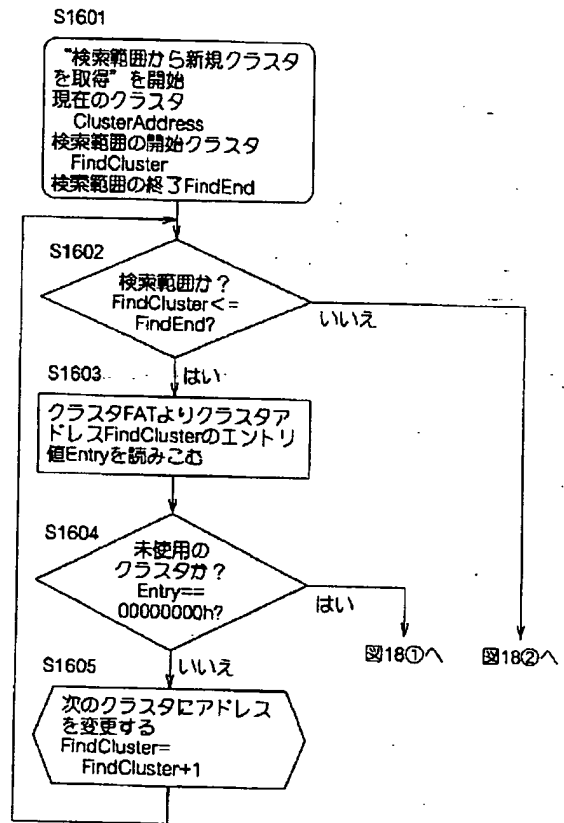
201	FILE1.DAT	202	2	203	1254
204	FILE2.DAT	205	4	206	2013
207	FILE3.DAT	208	8	209	47



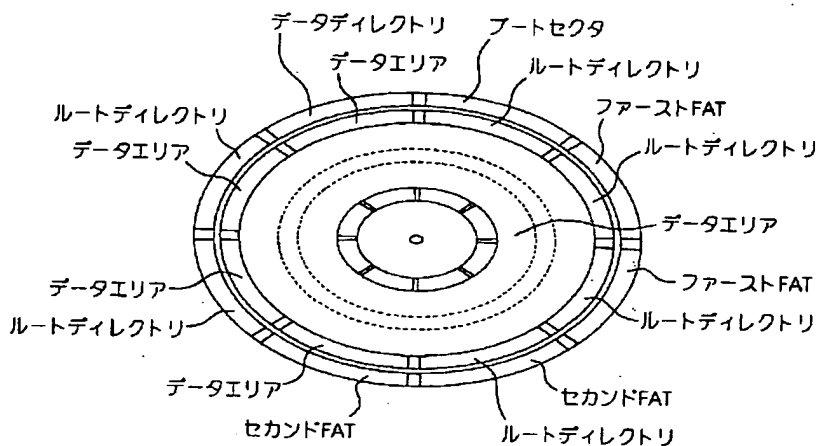
【図16】



【図17】



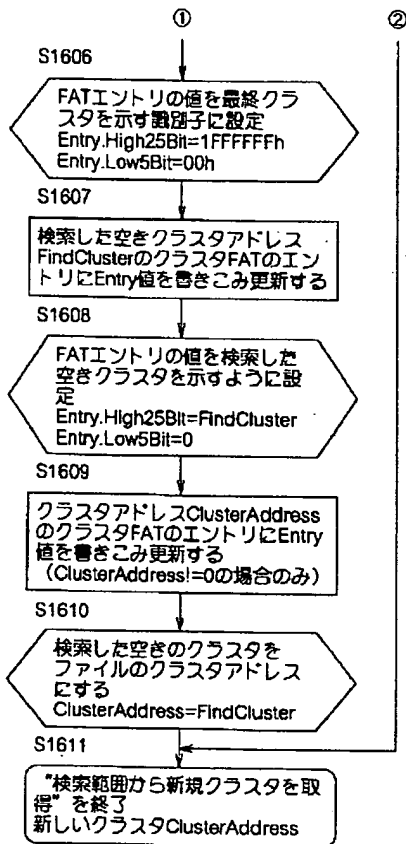
【図33】



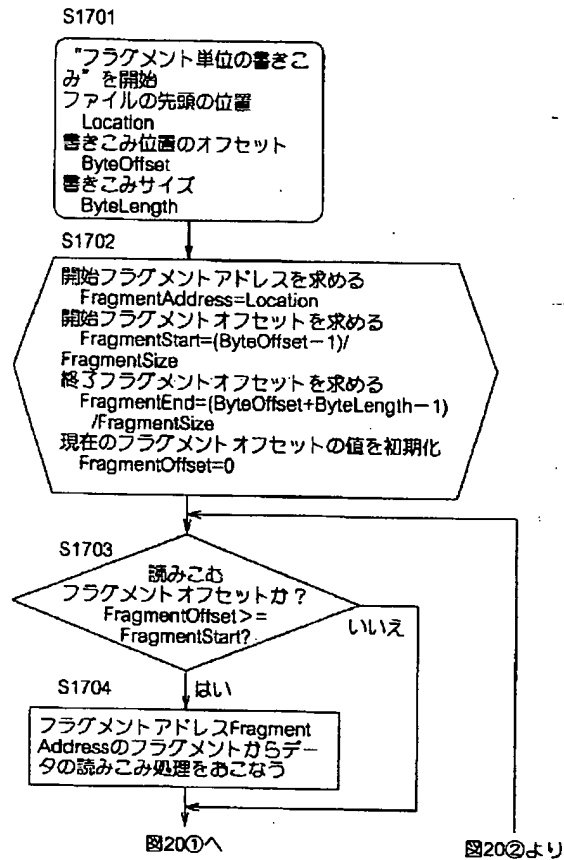
【図37】

401	4	402	T*2	403	7	404	T*3	405	11	406	T*1	407	9
-----	---	-----	-----	-----	---	-----	-----	-----	----	-----	-----	-----	---

【図18】



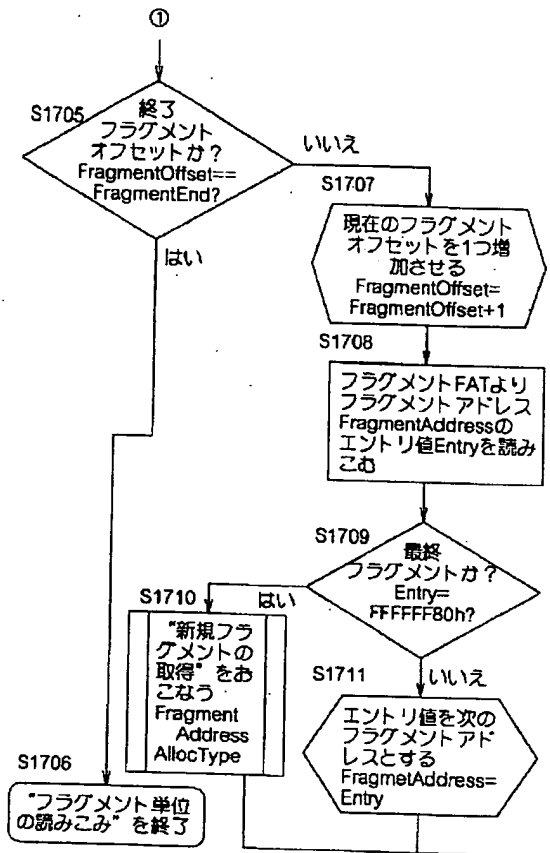
【図19】



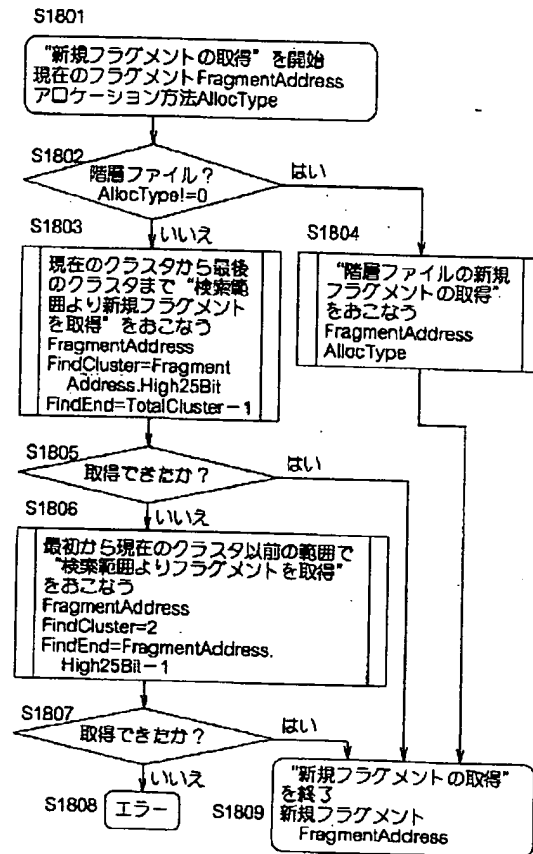
【図36】

301	302	303 <sub>(2)</sub> 3	304 <sub>(3)</sub> 6
305 <sub>(4)</sub> 7	306 <sub>(5)</sub> 0	307 <sub>(6)</sub> 65535	308 <sub>(7)</sub> 11
309 <sub>(8)</sub> 65535	310 <sub>(9)</sub> 65535	311 <sub>(10)</sub> 0	312 <sub>(11)</sub> 9
313 <sub>(12)</sub> 0			
		314 0	315 9

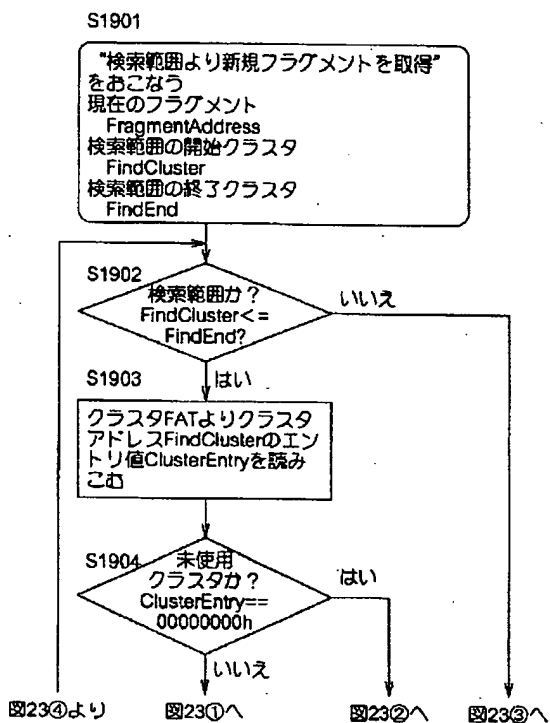
【図20】



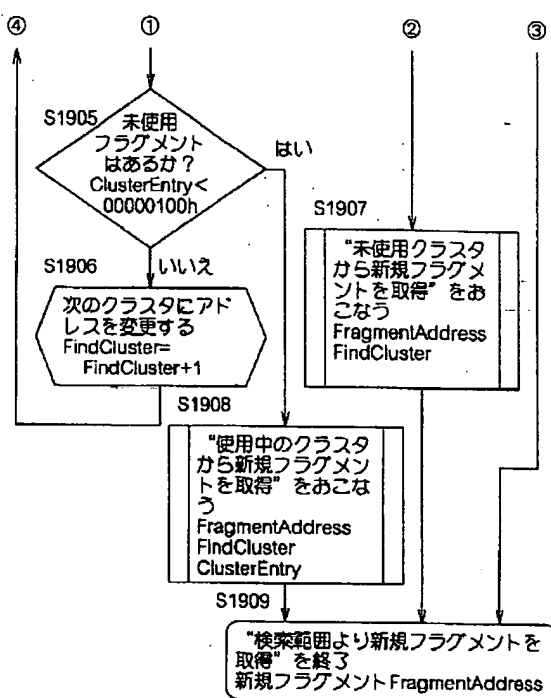
【図21】



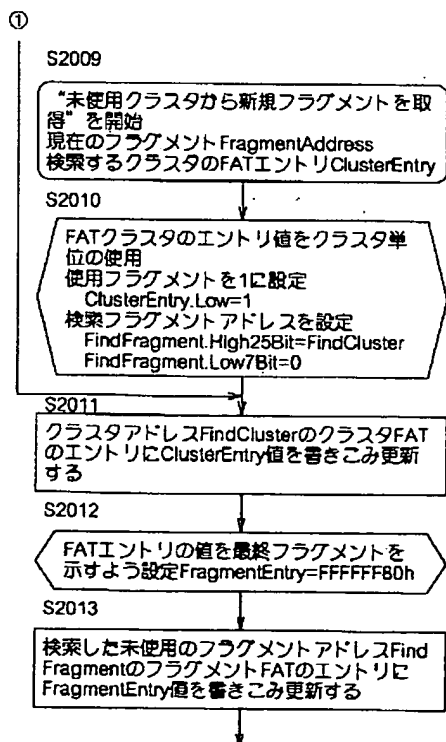
【図22】



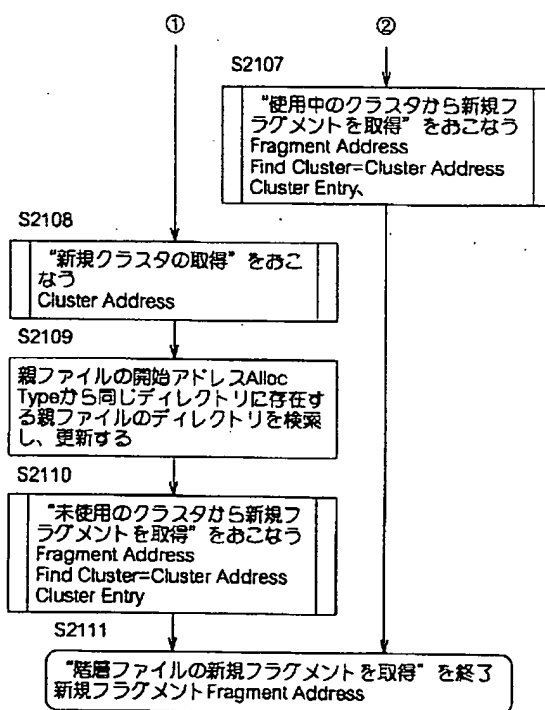
【図23】



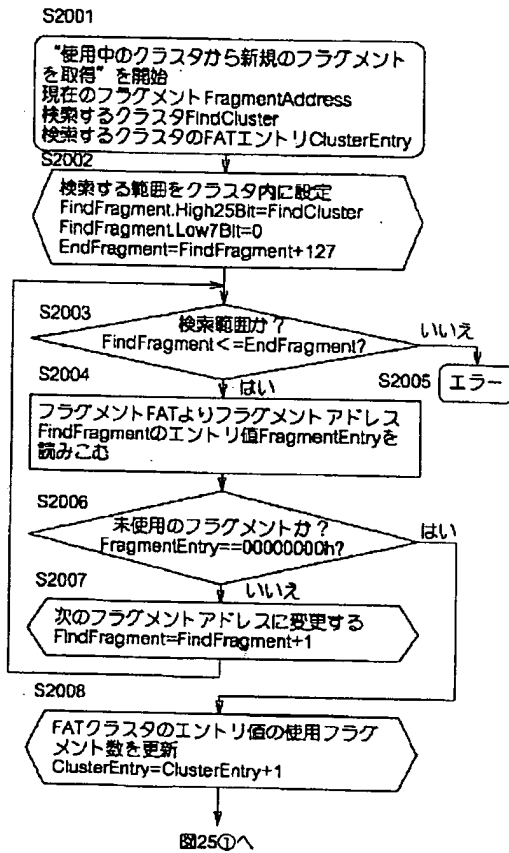
【図25】



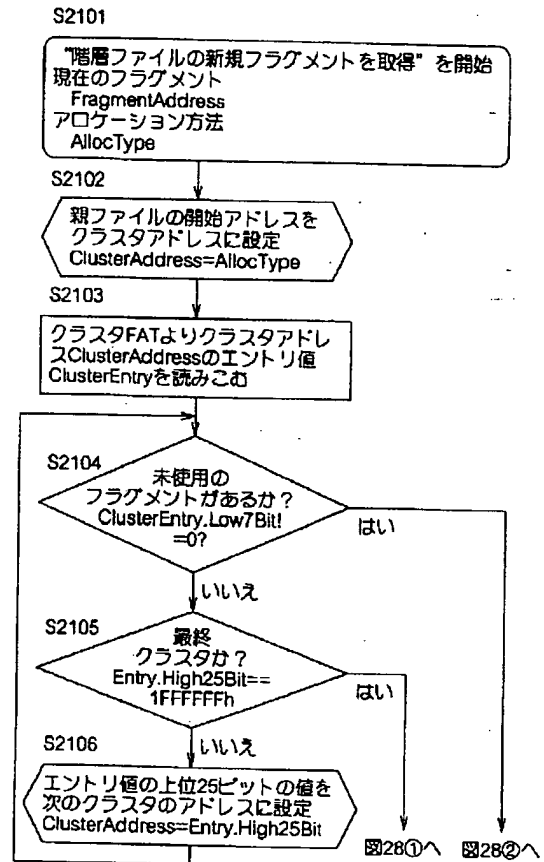
【図28】



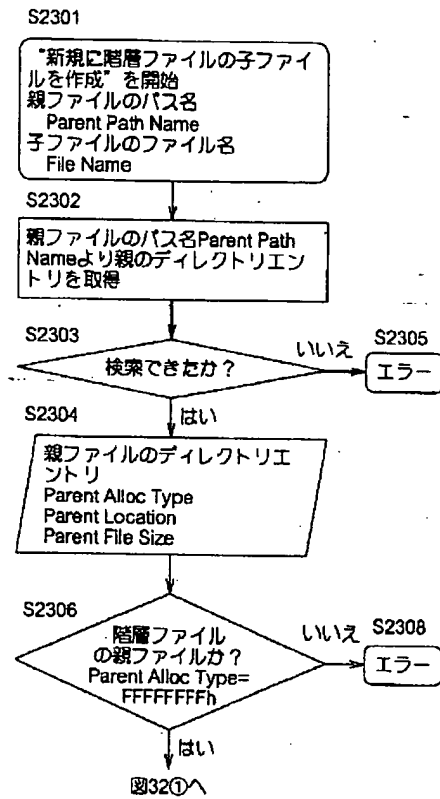
【図24】



【図27】



【図 3 1】



【図32】

